به نام خدا

کنترل سیستم های خطی

خانم دکتر مریم بابازاده

امیرحسین رستمی

96101635

پروژه ی اختیاری درس کنترل

هشدار:

جهت استفاده از فایل سیمولینک ابتدا Section اول فایل mlx. ضمیمه شده را ران کنید تا متغیر های ثابت استفاده شده در فایل سیمولینک به workspace متلب اضافه شود.

گام اول: استخراج مدل دینامیکی ball Segway:

a)برای اینکه ما به معادلات حالت برسیم در حالت غیر خطی کافی است که به صورت زیر عمل کنیم

$$X' = f(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m)$$

یعنی باید مشتق متغیرهای حالت را برحسب تابعی از متغیرهای حالت و ورودی ها نوشت حال کافی است که تنها یه تابع بین هردو پیدا کنیم و دیگر نیاز به یک ماتریس ضرایب عادی نداریم. متغیر های حالت ما در این حالت متغیر های $[y_k, \theta_x, y_k', \theta_x']$ می باشد که بدیهی است ما به راحتی میتوانیم در ماتریس حالتمان بگوییم که مشتق متغیر حالت اول و دوممان همان متغیر های حالت سوم و چهارممان میباشد، برای متغیر های سوم و چهارم ما از معادله اویلرـ لاگرانژ استفاده میکنیم،طبق معادله ی موجود در داک پروژه داریم که معادله ی دیفرانسیل فیزیک vector حالت qx به شرح زیر است:

$$q_x = [y_k, \theta_x] \rightarrow M(q_x)q_x'' + C(q_x, q_x')q_x' + D(q_x') + G(q_x) = Q_x(u_x)$$

که در اینجا M ماتریس جرم، G بردار جاذبه، D بردار اصطکاک Q بردار گشتاور کنترل در محور y و C هم ماتریس مربوط به Coriolis-centrifugal میباشد. این ماتریس ها به شرح زیر اند

$$\mathbf{M}(\mathbf{q}_x) = \begin{bmatrix} m_k + \frac{I_k}{r_k^2} + m_a + \frac{3I_w\cos^2\alpha}{2r_w^2} & \frac{3I_w\cos^2\alpha}{2r_w^2}r_k - m_a l\cos\theta_x \\ \frac{3I_w\cos^2\alpha}{2r_w^2}r_k - m_a l\cos\theta_x & m_a l^2 + \frac{3I_w r_k^2\cos^2\alpha}{2r_w^2} + I_x \end{bmatrix},$$

$$\mathbf{G}(\mathbf{q}_x) = [0 \quad -m_a g l\sin\theta_x]^T,$$

$$\mathbf{D}(\dot{\mathbf{q}}_x) = [b_y\dot{y}_k + d_y sign(\dot{y}_k) \quad b_{rx}\dot{\theta}_x + d_{rx} sign(\dot{\theta}_x)]^T,$$

$$\mathbf{Q}_x = \left[\frac{1}{r_w}\tau_x \quad \frac{r_k}{r_w}\tau_x\right]^T,$$

$$\mathbf{C}(\mathbf{q}_x, \dot{\mathbf{q}}_x) = \begin{bmatrix} 0 & m_a l\dot{\theta}_x\sin\theta_x \\ 0 & 0 \end{bmatrix},$$

از این معادله به راحتی میتوانیم q_x'' را برحسب مشتقات درجه پایین تر خود که متغیر های حالت ما می باشد کافی است که ترم مشتق دوم qx را به سمت دیگر معادله ببریم و الباقی را به سمت دیگر و وارون ماتریس M را در طرفین ضرب کنیم داریم که:

$$M(q_x)q_x'' = Q_x(u_x) - \left(C(q_x, q_x')q_x' + D(q_x') + G(q_x)\right) \rightarrow$$
$$q_x'' = M^{-1}(q_x)(Q_x(u_x) - \left(C(q_x, q_x')q_x' + D(q_x') + G(q_x)\right))$$

پس ما سطر سوم و چهارم ماتریس حالتمان را نیز پیدا کرده ایم در نتیجه به معادلات حالت زیر میرسیم:

$$\dot{\mathbf{x}}_r = \begin{bmatrix} \dot{\mathbf{q}}_x \\ \mathbf{M}^{-1}(\mathbf{q}_x)[\mathbf{Q}(u_x) - \mathbf{C}(\mathbf{q}_x, \dot{\mathbf{q}}_x)\dot{\mathbf{q}}_x - \mathbf{D}(\dot{\mathbf{q}}_x) - \mathbf{G}(\mathbf{q}_x)] \end{bmatrix}$$

$$\mathbf{F}_r(\mathbf{x}_r, u_x) = \begin{bmatrix} \dot{y}_k \\ \dot{\theta}_x \\ f_{r3}(\mathbf{x}_r) + g_{r3}(\mathbf{x}_r)u_x \\ f_{r4}(\mathbf{x}_r) + g_{r4}(\mathbf{x}_r)u_x \end{bmatrix},$$

که x_r همان بردار متغیر های حالتمان در فضای y-z میباشد یعنی $x_r = [y_k, \theta_k, y_k', \theta_k']^T$ و تنها در ماتریس Q ما اثری از ورودی یعنی گشتاور در راستای x (تاثیر گذار در صفحه y-z) را مشاهده می کنیم.

اگر مقادیر مطرح شده در بالا برای ماتریس های M,C,D,G را طبق ماتریس های مطرحی در بالا قرار بدهیم خواهیم داشت که پس از ساده سازی به ماتریس های زیر خواهیم رسید:

$$f_{r3}(\mathbf{x}_r) = -0.05\begin{pmatrix} \dfrac{7.59\sin\theta_x - 65.5\sin 2\theta_x + 11.5\dot{\theta}_x^2\sin\theta_x}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \\ +\dfrac{0.45b_y\dot{y}_k - 0.03b_{rx}\dot{\theta}_x + 0.52b_{rx}\dot{\theta}_x\cos\theta_x}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \\ +\dfrac{0.45d_y sign(\dot{y}_k) - 0.03d_{rx} sign(\dot{\theta}_x)}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \\ +\dfrac{0.52d_{rx} sign(\dot{\theta}_x)\cos\theta_x}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \end{pmatrix}, \qquad f_{r4}(\mathbf{x}_r) = \begin{pmatrix} \dfrac{25.1\sin\theta_x + 0.037\dot{\theta}_x^2\sin\theta_x - 0.32\theta_x^2\sin 2\theta_x}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \\ -\dfrac{0.1b_{rx}\dot{\theta}_x - 0.0015b_y\dot{y}_k + 0.025b_y\dot{y}_k\cos\theta_x}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \\ -\dfrac{0.1d_{rx} sign(\dot{\theta}_x) - 0.0015d_y sign(\dot{y}_k)}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \\ +\dfrac{0.025d_y sign(\dot{y}_k)\cos\theta_x}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \end{pmatrix},$$

$$g_{r3}(\mathbf{x}_r) = \frac{0.06\cos\theta_x + 0.21}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2}, \qquad g_{r4}(\mathbf{x}_r) = \frac{0.25\cos\theta_x + 0.21}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2};$$

توجه کنید که ux همان ورودی سیستم ما در این حالت و برابر با گشتاور در راستای x است.

B)در رابطه با صفحه y-z ما روابط مشابه با حالت قبل داریم، ایندفعه بردار حالت انتخاب شده به صورت $[x_k, \theta_y, x'_k, \theta'_y]$ میباشد و با استفاده از معادله اویلر لاگرانژ به طور مشابه داریم:

$$q_y = [x_k, \theta_y] \rightarrow M(q_y)q''_y + C(q_y, q'_y)q'_y + D(q'_y) + G(q_y) = Q_y(u_y)$$

که مشابه به قسمت قبل میتوانیم انرا ساده کنیم و به فرم زیر انرا بنویسیم

$$q''_y = M^{-1}(q_y)(Q_y(u_y) - \left(C(q_y, q'_y)q'_y + D(q'_y) + G(q_y)\right))$$

حال داریم که به کمک رابطه ی به دست آمده داریم که معادله حالت در این قسمت به شرح زیر می گردد:

$$\begin{aligned}\dot{\mathbf{x}}_p &= \begin{bmatrix}\dot{\mathbf{q}}_y \\ \mathbf{M}^{-1}(\mathbf{q}_y)[\mathbf{Q}(u_y) - \mathbf{C}(\mathbf{q}_y, \dot{\mathbf{q}}_y)\dot{\mathbf{q}}_y - \mathbf{D}(\dot{\mathbf{q}}_y) - \mathbf{G}(\mathbf{q}_y)]\end{bmatrix} \\ &= \mathbf{F}_p(\mathbf{x}_p, u_y),\end{aligned}$$

که ماتریس های M,Q,C,D,G به شرح زیراند:

$$\begin{aligned}
\mathbf{q}_y &= [x_k \ \ \theta_y]^T, \\
\mathbf{M}(\mathbf{q}_y) &= \begin{bmatrix} m_k + \dfrac{I_k}{r_k^2} + \dfrac{3I_w\cos^2\alpha}{2r_w^2} + m_a & m_a l\cos\theta_y - \dfrac{3I_w\cos^2\alpha}{2r_w^2}r_k \\ m_a l\cos\theta_y - \dfrac{3I_w\cos^2\alpha}{2r_w^2}r_k & \dfrac{3I_w r_k^2\cos^2\alpha}{2r_w^2} + m_a l^2 + I_y \end{bmatrix}, \\
\mathbf{G}(\mathbf{q}_y) &= [0 \ \ -m_a g l\sin\theta_y]^T, \\
\mathbf{D}(\dot{\mathbf{q}}_y) &= [b_x\dot{x}_k + d_x sign(\dot{x}_k) \ \ b_{ry}\dot{\theta}_y + d_{ry} sign(\dot{\theta}_y)]^T, \\
\mathbf{Q}_y &= \left[-\frac{1}{r_w}\tau_y \ \ \frac{r_k}{r_w}\tau_y\right]^T, \\
\mathbf{C}(\mathbf{q}_y, \dot{\mathbf{q}}_y) &= \begin{bmatrix} 0 & -m_a l\dot{\theta}_y\sin\theta_y \\ 0 & 0 \end{bmatrix},
\end{aligned}$$

که xp همانند قسمت قبل که xr معرف متغیر های حالت بود،در این حالت نیز معرف متغیر های حالت معرفی شده توسط ما است.اگر مقادیر ماتریسی عبارت های بالا را قرار بدهیم به معادله ی زیر می رسیم:

$$\mathbf{F}_p(\mathbf{x}_p, u_y) = \begin{bmatrix} \dot{x}_k \\ \dot{\theta}_y \\ f_{p3}(\mathbf{x}_p) + g_{p3}(\mathbf{x}_p)u_y \\ f_{p4}(\mathbf{x}_p) + g_{p4}(\mathbf{x}_p)u_y \end{bmatrix}.$$

که توابع موجود در سطر سوم و چهارم به شرح زیر اند و uy گشتاور در راستای y است که در اصل برابر است با:TauY داریم که

$$f_{p3}(\mathbf{x}_p) = -0.05\left(\begin{aligned} &\frac{65.5\sin 2\theta_y - 7.59\sin\theta_y - 11.3\dot{\theta}_y^2\sin\theta_y}{0.075\cos\theta_y - 0.65\cos\theta_y + 2.2} \\ &+\frac{0.03b_{ry}\dot{\theta}_y + 0.44b_x\dot{y}_k - 0.52b_{ry}\dot{\theta}_y\cos\theta_y}{0.075\cos\theta_y - 0.65\cos\theta_y + 2.2} \\ &+\frac{0.03d_{ry}sign(\dot{\theta}_y) + 0.44d_x sign(\dot{x}_k)}{0.075\cos\theta_y - 0.65\cos\theta_y + 2.2} \\ &-\frac{0.52d_{ry}sign(\dot{\theta}_y)\cos\theta_y}{0.075\cos\theta_y - 0.65\cos\theta_y + 2.2}\end{aligned}\right),$$

$$f_{p4}(\mathbf{x}_p) = -\left(\begin{aligned} &\frac{-25.1\sin\theta_y - 0.037\dot{\theta}_y^2\sin\theta_y + 0.32\dot{\theta}_y^2\sin 2\theta_y}{0.075\cos\theta_y - 0.65\cos^2\theta_y + 2.2} \\ &+\frac{0.1b_{ry}\dot{\theta}_y + 0.0015b_x\dot{x}_k - 0.025b_x\dot{x}_k\cos\theta_y}{0.075\cos\theta_y - 0.65\cos^2\theta_y + 2.2} \\ &+\frac{0.1d_{ry}sign(\dot{\theta}_y) + 0.0015d_x sign(\dot{x}_k)}{0.075\cos\theta_y - 0.65\cos^2\theta_y + 2.2} \\ &-\frac{0.025d_x sign(\dot{x}_k)\cos\theta_y}{0.075\cos\theta_y - 0.65\cos^2\theta_y + 2.2}\end{aligned}\right),$$

$$g_{p3}(\mathbf{x}_p) = -\frac{0.06\cos\theta_y + 0.21}{0.075\cos\theta_y - 0.65\cos^2\theta_y + 2.2}, \qquad g_{p4}(\mathbf{x}_p) = \frac{0.25\cos\theta_y + 0.21}{0.075\cos\theta_y - 0.65\cos^2\theta_y + 2.2}.$$

ایرادات ماتریس های بزرگ مقاله:

توجه کنید که پس از انجام محاسبات و ضرب سیمبولیک ماتریس ها داریم که اشکالاتی در ماتریس های فوق وجود داشت که پس از مراجعه به استاد درس و بررسی صحت وارد بودن ایرادات به این نتیجه رسیدیم که ایراداتی نگارشی در معادله رخ داده است که به شرح زیر اند:

1ـ تمامی مخرج های تک تک سطر های fp3 عبارت مخرج عبارت زیر است:

$$0.075\cos\theta_y - 0.65\cos^2\theta_y + 2.2$$

و سهوا در برخی کسر ها توان دو کسینوس(عبارت دوم در مخرج) حذف شده است.

2ـ خط دوم fp3 در صورت xkDot است و سهوا به صورت ykDot نوشته شده است.

c)طبق متن مقاله برای رابطه θ_z'' رابطه زیر را داریم:

$$\ddot{\theta}_z = \frac{I_k}{I_k I_z + 3(I_k + I_z) I_w \frac{r_k^2}{r_w^2} \sin^2 \alpha} \frac{r_k}{r_w} \tau_z,$$

که مقدار ضریب تقریبا برابر تقریبا 0.17 می باشد پس معادلات حالت به صورت زیر میباشد:

$$[\theta_z \;\; \theta_z']' = [\theta_z' \quad 0.16806 u_z]$$

توجه که u_z گشتاور ورودی در راستای z است که همان TauZ است.

گام دوم: ساخت مدل غیر خطی Ball Segway در سیمولینک:

بنده به دو روش به پیاده سازی این بخش های پرداختم**(هر دو روش پیاده سازی در کد موجود است):**

1ـ پیاده سازی ماتریس ها به صورت سمبولیک و ضرب آن ها در یکدیگر

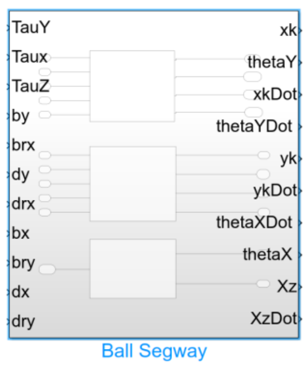
2ـ نوشتن ریز نتایج ضرب ماتریس های مقاله(ماتریس های بزرگ!!) و تعریف سیمبولیکی آن ها

توضیحی راجع به تک تک بخش های این قسمت:

در تک تک قسمت ها ابتدا معادله ی دیفرانسیل موجود را پیاده سازی کرده سپس مشتق دوم تولید شده را از 2 انتگرال گیر عبور می دهم تا به مشتق اول و خود آن عبارت برسم و در نهایت سیم کشی های لازم را انجام داده و خروجی ها به شرح زیر گردید.

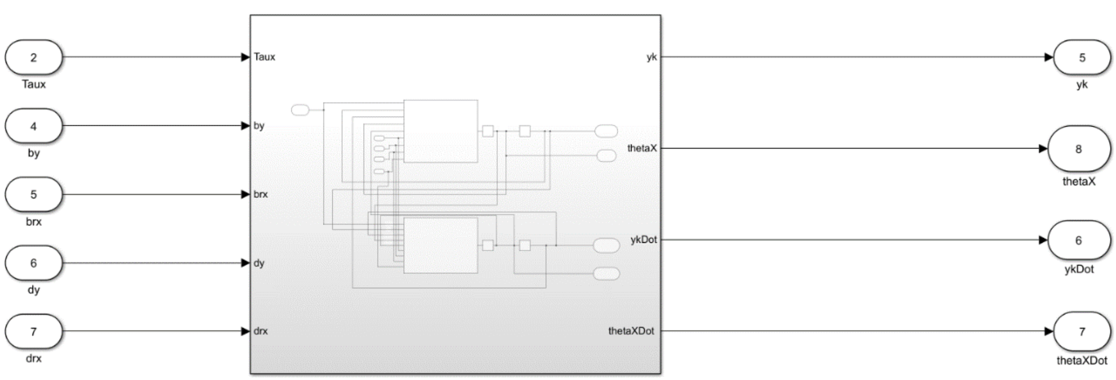
(A, B, C

در نهایت کل سیمولینک بخش ball Segway به شکل زیر است:

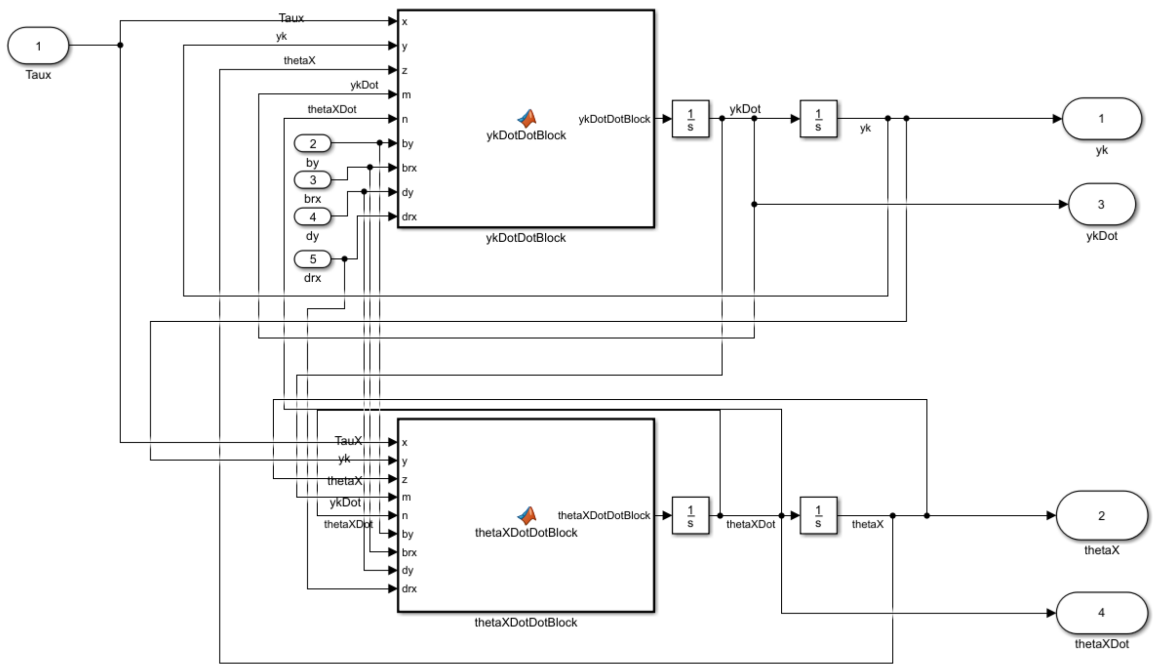


حال با کلیک روی کل بلوک به بخش زیر می رسیم:

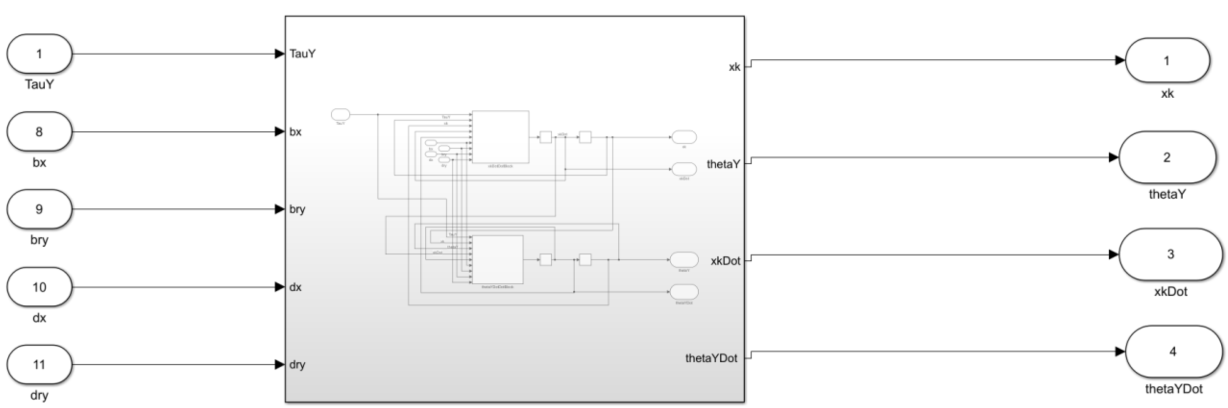
بخش اول:



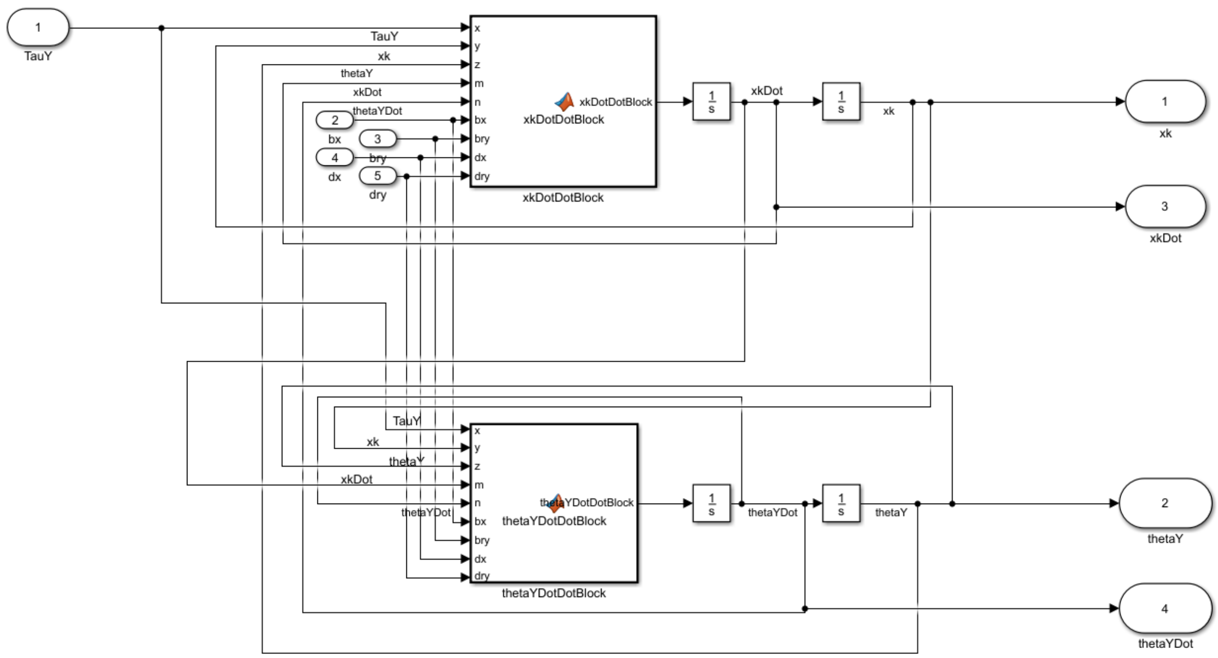
حال با کلیک روی این بلوک داریم که به شکل سیمولینکی نهایی زیر می رسیم:



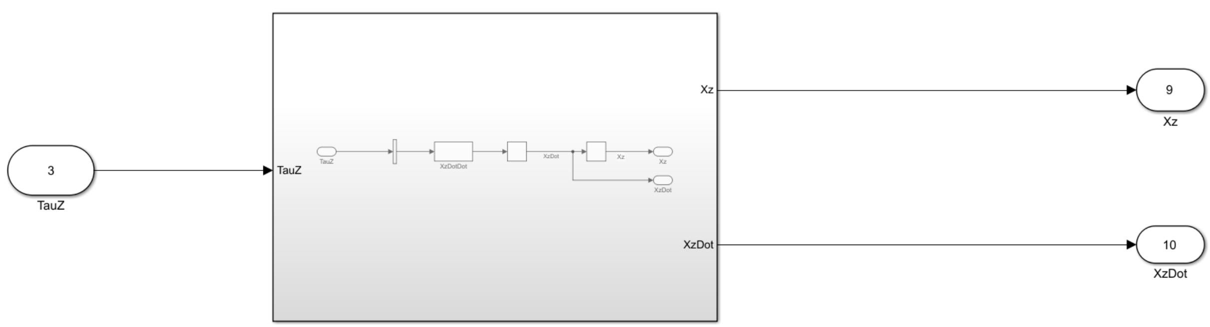
بخش دوم:



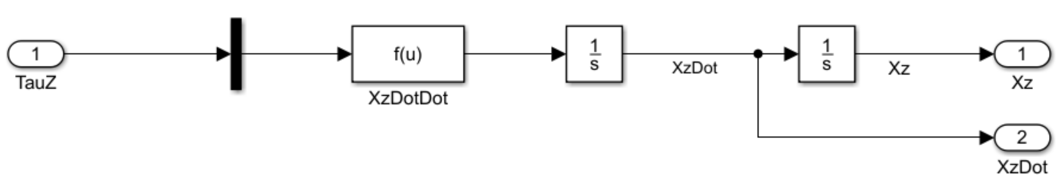
حال با کلیک روی این بلوک به نهایی ترین پیاده سازی زیر می رسیم:



بخش سوم:

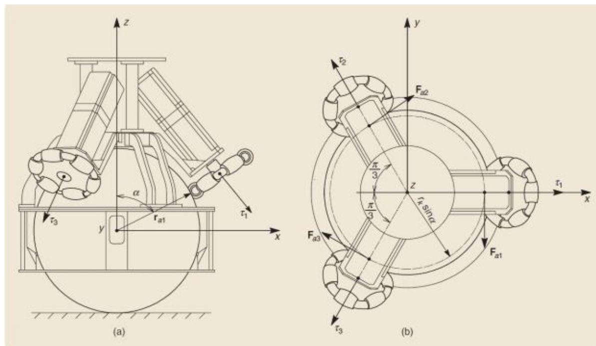


حال با کلیک روی این بلوک داریم که به شکل سیمولینکی نهایی زیر می رسیم:



گام سوم:

در مقاله برای راحت تر شدن معادلات کنترلر ها و متغیر حالت های ballSegway در نظر گرفته شده که نهایتا گشتاور هایی در راستا های x,y,z به سیستم اعمال می شود اما در حقیقت داریم که موتور های متصل به بدنه ی ballSegway در راستاهایی غیر از x,y,z است و همانطور که در شکل زیر ملاحظه می کنید راستای موتور های متصل به بدنه در راستای غیر از x,y,z است و لذا طبیعی است که مقدار آن ها با مقدار گشتاور موجود در راستاهای x,y,z نابرابر باشد.به شکل زیر توجه کنید:



با توجه به متن مقاله گشتاور های غیرحقیقی x,y,z، گشتاور های **مجازی**(virstual) و گشتاور موتورها،گشتاور های **حقیقی** نام گذاری شده است.توجه کنید که خب طبیعا خروجی کنترلر ها باید از جنس گشتاور های virtual باشد لذا لازم است که گشتاورهای x,y,z را به گشتاور هایی در راستای موتورک های حول گوی درنظر بگیریم و ماتریسی که این تبدیل را انجام بدهد به شرح زیر است:

$$\begin{bmatrix}\tau_1\\ \tau_2\\ \tau_3\end{bmatrix}=\begin{bmatrix}\frac{2}{3\cos\alpha} & 0 & \frac{1}{3\sin\alpha}\\ -\frac{1}{3\cos\alpha} & \frac{\sqrt{3}}{3\cos\alpha} & \frac{1}{3\sin\alpha}\\ -\frac{1}{3\cos\alpha} & -\frac{\sqrt{3}}{3\cos\alpha} & \frac{1}{3\sin\alpha}\end{bmatrix}\begin{bmatrix}\tau_x\\ \tau_y\\ \tau_z\end{bmatrix}.$$

توجه کنید که خروجی کنترلر ها به صورت گشتاور هایی در راستای x,y,z است ولی ورودی موتورهای ballSegway به صورت گشتاور های Tau1,Tau2,Tau3 است لذا لازم است که خروجی کنترلر ها با ماتریس تبدیل فوق به به گشتاور هایی در راستای موتور تبدیل گردند اما در این پروژه چون ما صرفا شبیه سازی کرده ایم و نیاز نیست که این واحد در مسیر کنترلی قرار بگیرد اما بنده این واحد را جدا قرار داده ام که استاد عزیز در صورت لزوم خروجی آن هارا هم ملاحظه کند.

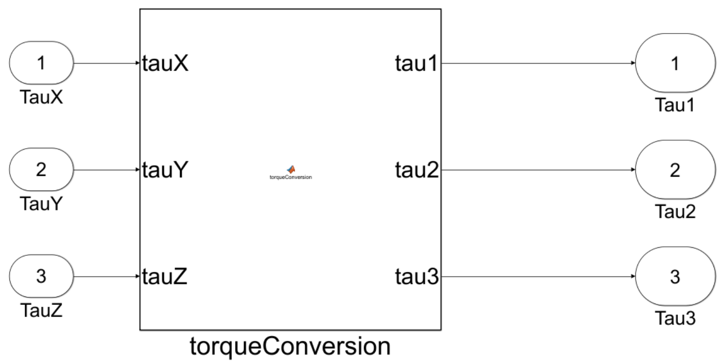
حال طبق ورودی قسمت 3 داریم که اگر بخواهیم از گشتاور در هر صفحه به گشتاور های واقعی برسیم با فرض الفا برابر 65.5 درجه باید گشتاور هارا در ماتریس زیر ضرب کنیم:

1.6076	0	0.3663
-0.8038	1.3922	0.3663
-0.8038	-1.3922	0.3663

و اگر بخواهیم که گشتاور های τ_1, τ_2, τ_3 را تبدیل به گشتاورد در صفحات کنیم باید ماتریس را معکوس کرده که مقادیر ان به صورت زیر میشود:

0.4147	-0.2073	-0.2073
0	0.3591	-0.3591
0.9100	0.9100	0.9100

از انجاکه ما در حال کار روی معادلات هستیم نیازی به τ_1, τ_2, τ_3 نداریم و خود کنترلر ها گشتاور ها را برحسب گشتاور در صفحات تولید می کنند.



گام چهارم:

تولید کنترلر چرخش:

در مقاله دو رابطه ی مرتبط با این بخش داده شده:

1-

$$\tau_z = K_{dz}\dot{\theta}_z + K_{pz}(\theta_z - \theta_{zd}).$$

2-

$$\ddot{\theta}_z = \frac{I_k}{I_k I_z + 3(I_k + I_z) I_w \frac{r_k^2}{r_w^2} \sin^2\alpha} \frac{r_k}{r_w} \tau_z,$$

که همانطور که ملاحظه کردید پس از محاسبه ی ضریب گشتاور TauZ داریم که این ضریب 0.168 است با تلفیق روابط فوق داریم که تابع تبدیل حلقه باز آن به صورت زیر است:

$$G(s) = -\frac{k_{pz}\beta}{s(s - k_{dz}\beta)} = \frac{\theta_z}{\theta_{zd}}$$

(توجه کنید به بتا همان ضریب 0.168 محاسبه شده در قسمت بالاست)

اگر با فرم معروف حلقه باز درجه دو آنرا مقایسه کنیم که به فرم زیر است:

$$\frac{w_n^2}{s(s+2\zeta w_n)}$$

با فرض اینکه جنس ورودی زاویه دلخواه از پله میباشد ما چون سیستم مرتبه یک است خطای ماندگار نداریم پس نگران این موضوع نیستیم حال با تلفیق روابط موجود داریم که مقادیر kpz و kdz به شرح زیر با مشخصات زتا و Wn ارتباط دارد:

مقادیر به صورت زیر بدست می آید :

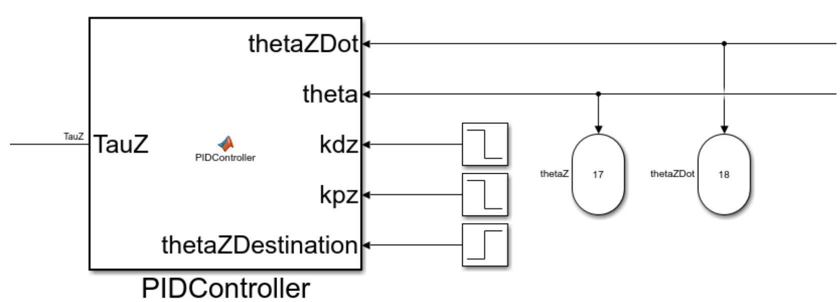
$$k_{pz}=-\frac{w_n^2}{\beta}, k_{dz}=-\frac{2\xi w_n}{\beta}$$

حال از انجا که اگر تغییرات θ_z شدید باشد و دیر میرا گردد برای کاربر بسیاااار نامطلوب می گردد لذا لازم است تا نرخ damping را زیاد درنظر بگیریم و حدودا مقدار 0.9 مناسب است و هم چنین اینکه Ball Segway هرچه زودتر به حالت پایدار خود برسد نیز برای ما مهم است(درنظر بگیرید که کاربر چقدر اذیت می شود که زمان های Ts , Tr زیاد باشد لذا از آنجا که Ts رابطه معکوس با ζw_n دارد هرچه w_n بزرگتر باشد برای ما بهتر است حدودا و انرا 5 در نظر میگیریم پس مقادیر برابر با حدودا :

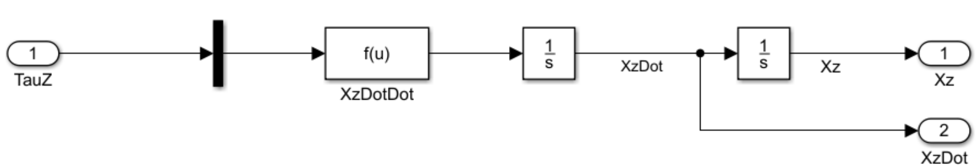
$$k_{pz}=-150, k_{dz}=-60$$

در نظر میگیریم.

b) ابتدا مدل سیمولینک این قسمت را شرح می دهیم:



کنترلر این بخش به شکل زیر است که مقادیر پورت های TauZ,thetaZDot,Theta به بلوک بخش زیر متصل می گردد:

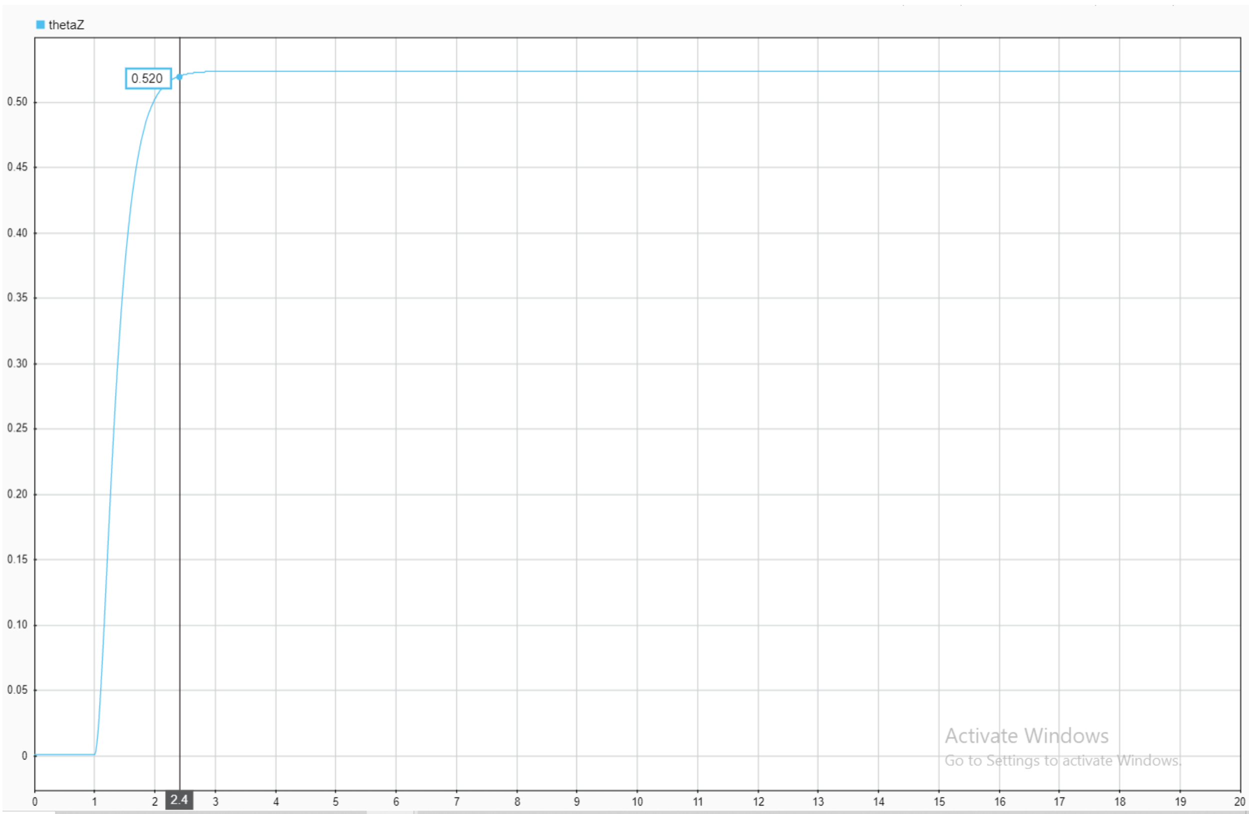


توجه : منظور از theta همان Xz و منظور از thetaZDot همان XzDot است.

حال برای تست کردن کیفیت عملکرد این بخش داریم که به thetaZD مقدار 30 درجه یا همان pi/6 را می دهیم.

نکته:توجه کنید که در فایل سیمولینک،ضرایب و مقدار thetaZD به صورت پله به کنترلر کننده ما داده می شود.حال برویم سراغ خروجی ها:

خروجی thetaZ:



که در حدود 1.4 ثانیه(**توجه کنید که من در ثانیه ی اول ورودی را به بلوک ها می دهم**) به مقدار مطلوب میرسد و مقدار خیلی کمی اورشوت دارد این یک طراحی واقعا ایده ال برای زاویه ما میباشد و به خوبی کار می کند!

خروجی thetaZDot: (توجه کنید که من در ثانیه ی اول ورودی را به بلوک ها می دهم)

خروجی TauZ: (توجه کنید که من در ثانیه ی اول ورودی را به بلوک ها می دهم)

بررسی اثر تغییرات زتا و Wn :

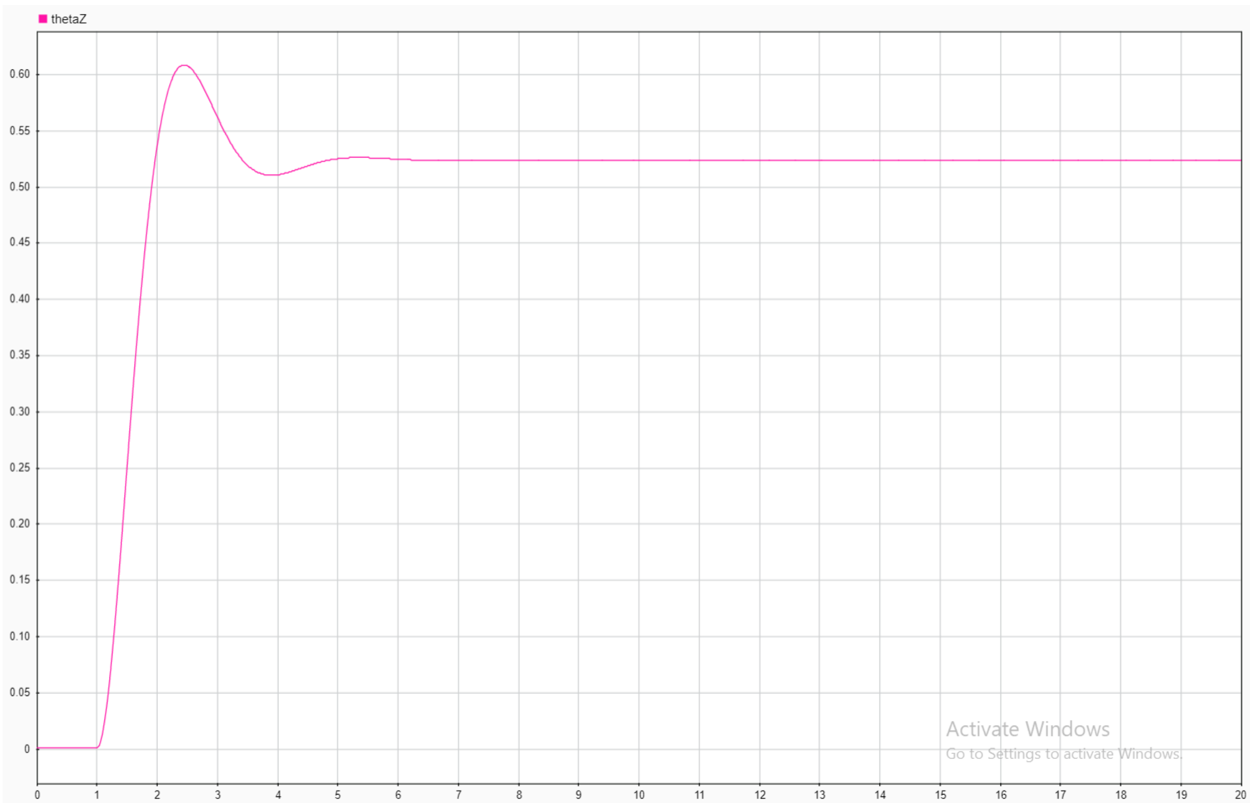
۱ ــ اثرتغییر زتا

اگر مثلا ما مقدار زتا را 0.5 انتخاب میکردیم(توجه کنید این مقدار برای thetaZ واقعا فاجعه است و صرفا ما می خوایم که اثرات تغییرات این مقادیر را در این بخش بررسی کنیم)لذا طبق روابط فوق مقدار ضریب تقریبا 30- = kdz می گردد و خروجی به شکل زیر می شود

که این با روابط همخوانی دارد! یعنی با کاهش damping ratio مقدار overshoot افزایش یافته است.

۲ـ اثر تغییر Wn

اگر wn را کمتر انتخاب کنیم دیرتر به مقدار نهایی خود میرسد برای این امر wn را نصف می‌کنیم و زتا را برابر حالت قبل در نظر میگیریم و داریم که خروجی آن به شرح زیر می‌گردد:



که همانطور که مشاهده می‌کنید کاهش Wn باعث کند شدن رسیدن به مقصد سیستم ما می‌گردد.

در ادامه راجع به اهمیت وجود این کنترلر صحبت می‌کنیم.

اهمیت وجود کنترلر:(دقت کنید به علامت دو تا k به شدت مهم است و اگر مثبت باشد شدیدا سامانه ناپایدار می گردد).

در مورد اهمیت کنترلر اگر کنترلر وجود نباشد که بدیهی است رابطه ناپایدار است جون رابطه در مخرج s^2 دارد که مطلوب نمیباشد در حلقه بسته با کنترلر نیز ما اگر مقادیر k ها منفی نباشد ناپایداری داریم، این از رابطه نیز معلوم میباشد اگر به ان دقت کنیم داریم:

$$\tau_z = K_{dz}\dot{\theta}_z + K_{pz}(\theta_z - \theta_{zd}).$$

اگر k_{dz} مثبت باشد یعنی گشتاور با افزایش سرعت بیشتر میشود که این یک فیدبک مثبت و هیچگاه سیستم نمی ایستد! و برای kpz اگر تتا بیشتر از تتای مطلوب باشد باید گشتاور منفی اعمال کند پس علامت هردو باید منفی باشد، وگرنه ناپایدار و مدام فاز تغییر می کند مثلا اگر مقادیر این دو تا k را مقادیری مثبت بدهیم(هرچه مثبت تر واگراتر می گردد خروجی)خروجی به عنوان مثال به صورت زیر می گردد:

که به خوبی ناپایداری مشخص است، یعنی مدام سیستم به طور خود می چرخد و به حالت ثابتی نمیرسد! ولی اگر پایداری را درست کنیم و با فرض جنس زاویه ورودی از نوع پله میباشد این مشکل وجود **ندارد** و به خاطر اینکه سیستم **مرتبه یک** میباشد بدون خطای ماندگار حالت سیستم به مقدار مطلوب می رسد و با تعیین پارامتر ها میتوان مقدار overshoot و سرعت رسیدن به حالت پایدار را درست کرد.

گام پنجم طراحی کنترلر های تعادلی:

بخش اول:هدف های اصلی کنترلر ما، متعادل کردن و حفظ لختی در برابر تغییرات ناگهانی شرایط اولیه توپ است.

ما در اینجا 2 تا کنترلر داریم یکی کنترلر PI که لوپ داخلی ما محسوب می شود و یک کنترلر LQR که جزو بخش کنترلر در حلقه بیرونی است(این شیوه طراحی مزیت سریع بودن را به ما می دهد، زیرا حلقه داخلی به سرعت بیشتری نسبت به حلقه بیرونی به روز رسانی میشود و سریع تر عمل میکند و این برتری آن نسبت به یک کنترلر تک حلقه که به تبع باید ساختار پیچیده ای داشته باشد دارد(این عبارت گفته ی خانم دکتر بابازاده بود در جلسه ای که خدمت ایشون رفتیم.)

هدف های بالا را به صورت اینکه θ_x, θ_y حول صفر پایدار شده باشند و x_k, y_k مقصد مورد نظر را دنبال کند.با توجه به گفته های مقاله می توانیم وظیفه ی کنترلر را کلا در دو بخش

Balancing Controller: هدف کنترلر تعادلی ، بالانس کردن زاویه های بدنه در زاویه دلخواه می باشد، که برای ما صفر میباشد. LQR که کنترلر حلقه بیرونی ما میباشد سرعت مرجع برای کنترلر PI مارا تولید میکند. وقتی که این دو در حال اجرای عملیات متعادل سازی اند، سرعت مرجع PI حول صفر توسط کنترلر LQR بدست میاید. در این حال PI توپ را در مختصات افقی نگه می دارد.

Transferring Controller: کنترلر تعادلی به ما این کمک را کرد تا اینکه در زاویه های مورد نظر در تعادل قرار بگیریم ولی توسط آن نمی توانیم توپ را به مکان دلخواه ببریم یا سرعت دلخواه را کسب کنیم. برای همین باید یک کنترلر که حرکت توپ را در اختیار داشته باشد. اگر کنترلر انتقالی را در نظر بگیرید، LQR نقش کنترلر سرعت را در اختیار دارد، خروجی های این کنترلر مکان و سرعت مرجع میباشند، براساس خطای بین سرعت حال و سرعت مورد نظر.LQR دو مورد مهم را کنترل میکند

1ـ متوقف کردن کنترلی که کند شدن و متقف شدن وقتی تحت تاثیر یک اغتشاش بیرونی بزرگ قرار میگیرد دارد.

2ـ فراهم اوردن سرعت توسط کاربر با joystick های تعبیه شده است. که کاربر میتواند سرعت را با این تنظیم کند، سیگنال کنترلر جوی استیک ها به صورت شیب(ramp) میباشند تا تغییر زاویه های بدنه را کاهش بدهد.

قسمت مربوط به مقاله راجع به این کنترلر ها و اهمیت وجودی آن ها:(کنترلر های LQR و PI)

Transferring Controller

The aforementioned balancing controller is good at balancing with the desired body angles; however, the ball segway cannot move to any desired ball position nor gain speed on the floor with the balancing controller. Therefore, a transferring controller is designed to control the movement of the ball segway.

Considering the transferring controller, the LQR outer-loop controller serves as a speed controller. The outputs of the LQR outer-loop controller are position and speed references, depending on the errors between the current and desired speeds of the ball and between the current and desired positions of the ball. The LQR outer-loop controller has two major applications. The first is a stopping controller that enables the ball segway to slow down and stop when subjected to large external disturbances. This application is utilized as a robust feature of the double-loop control system against external disturbances. The second is for movement where the user can provide velocity commands using a joystick. The control signal from the joystick is ramped to reduce the initial tilt angles of the body while controlling movement.

Balancing Controller

The objective of the balancing controller is to balance the body with the desired angles, which are zero for a pure balancing operation. The LQR outer-loop controller provides the speed reference to the PI inner-loop controller. When the LQR and PI double-loop controller executes a balancing function, the speed reference of the PI inner-loop controller is maintained around zero by the LQR outer-loop controller. Meanwhile, the PI inner-loop controller maintains the ball segway in a vertical position and fixes it on the floor.

Station-keeping is an effort by the ball segway to retain a specific position on the floor using the LQR outer-loop controller.

Robustness Controller

The PI inner-loop controller automatically compensates for various frictional torques for velocity tracking; thus, it reduces the effect of uncertainties of friction. In Figure 7, additional control torques u_{xf} and u_{yf} are considered as two feedforward controls that eliminate most uncertainties caused by viscous and Coulomb frictions as well as model and system parameters. The control gains of the PI inner-loop controller are empirically selected to provide proper performance for velocity tracking.

بخش دوم:

طبق پارامتر های بهینه معرفی شده را به راحتی برای کنترلر LQR پیاده سازی میکنیم، برای طراحی کنترلر PI باید مراحل زیر را طی کنیم(تک تک این قسمت ها طبق ترتیب مقاله گزارش داده شده است)

$$u_x = u_{xf} + K_{pr}(\nu_y - \dot{y}_k) + K_{ir}(x_{5r} - y_k),$$
$$u_y = u_{yf} + K_{pp}(\nu_x - \dot{x}_k) + K_{ip}(x_{5p} - x_k),$$

که v ها سرعت مطلوب تولید شده میباشند و x5r انتگرال از vy و x5p انگرال از vx میباشد. عبارت های uxf ,uyf نیز ناشی از کنترلر feedforward ما میباشند. هدف نهایی کنترلر رسیدن به سرعت مطول میباشد پس ما خطای سرعت را به صورت زیر تعریف میکنیم:

$$e_{vy}(t) = v_y - \dot{y}_k.$$

برحسب t از ان مشتق میگیریم پس داریم

$$\dot{e}_{vy}(t) = \dot{v}_y - \ddot{y}_k = \dot{v}_y - [f_{r3}(\mathbf{x}_r) + g_{r3}(\mathbf{x}_r)u_x].$$

فرض کنید داریم:

$$u_x = g_{r3}^{-1}(\mathbf{x}_r)\Big[-f_{r3}(\mathbf{x}_r) + \dot{v}_y + K'_{pr}e_{vy} + K'_{ir}\int_0^t e_{vy}(\tau)d\tau\Big].$$

با جاگذاری در معادله داریم :

$$\dot{e}_{vy} = -K'_{pr}e_{vy} - K'_{ir}\int_0^t e_{vy}(\tau)d\tau.$$

مشتق گیری برحسب زمان به ما میدهد:

$$\ddot{e}_{vy} + K'_{pr}\dot{e}_{vy} + K'_{ir}e_{vy} = 0.$$

برای اینکه سیستم را پایدار کنیم و همزمان انتقال هم پیدا کنیم، ارور حلقه بسته را به فرم معادله مشخصه درجه دوم استاندارد تبدیل میکنیم:

$$s^2 + K'_{pr}s + K'_{ir} = s^2 + 2\xi\omega_n s + \omega_n^2,$$

که جفت عبارت کسی و زتا باید بزرگ تر از صفر باشند. با جاگزاری معادلات داریم:

$$u_x = -g_{r3}^{-1}(\mathbf{x}_r) f_{r3}(\mathbf{x}_r) + g_{r3}^{-1}(\mathbf{x}_r)\dot{v}_y + 2g_{r3}^{-1}(\mathbf{x}_r)\xi\omega_n e_{vy} + g_{r3}^{-1}(\mathbf{x}_r)\omega_n^2 \int_0^t e_{vy}(\tau)d\tau.$$

در خطی سازی نقطه تعادل مشتق vy برابر صفره پس عبارت ها در نقطه صفر بردار تعادل حفظ میشود. پس کنترلر به صورت زیر بیان میشود:

$$\begin{aligned} u_x &= -g_{r3}^{-1}(\mathbf{0}) f_{r3}(\mathbf{0}) + 2g_{r3}^{-1}(\mathbf{0})\xi\omega_n e_{vy} + g_{r3}^{-1}(\mathbf{0})\omega_n^2 \int_0^t e_{vy}(\tau)d\tau \\ &= u_{xf} + K_{pr} e_{vy} + K_{ir} \int_0^t e_{vy}(\tau)d\tau, \end{aligned}$$

پس پارامتر ها به صورت زیر معرفی میشود:

$$\begin{aligned} K_{pr} &= 2g_{r3}^{-1}(\mathbf{0})\xi\omega_n, \\ K_{ir} &= g_{r3}^{-1}(\mathbf{0})\omega_n^2, \\ u_{xf} &= -g_{r3}^{-1}(\mathbf{0}) f_{r3}(\mathbf{0}). \end{aligned}$$

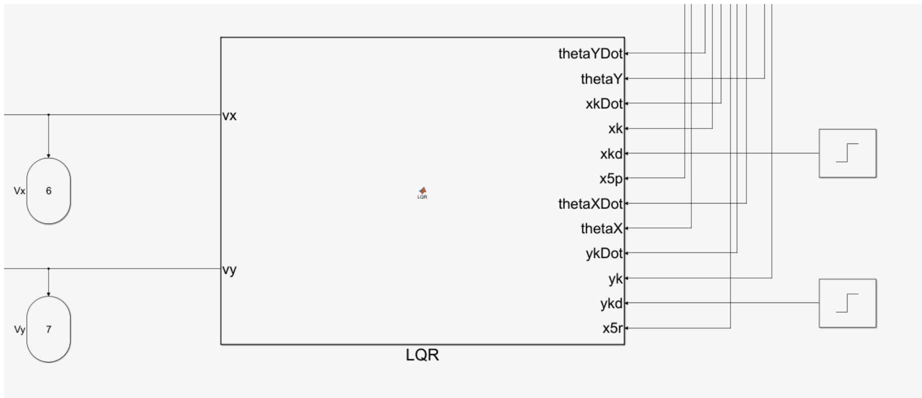
به طور مشابه برای صفحه x-z را محاسبه کرده و نتیجه به صورت زیر میشود:

$$\begin{aligned} K_{pp} &= 2g_{p3}^{-1}(0)\xi\omega_n, \\ K_{ip} &= g_{p3}^{-1}(0)\omega_n^2, \\ u_{yf} &= -g_{p3}^{-1}(0) f_{p3}(0). \end{aligned}$$

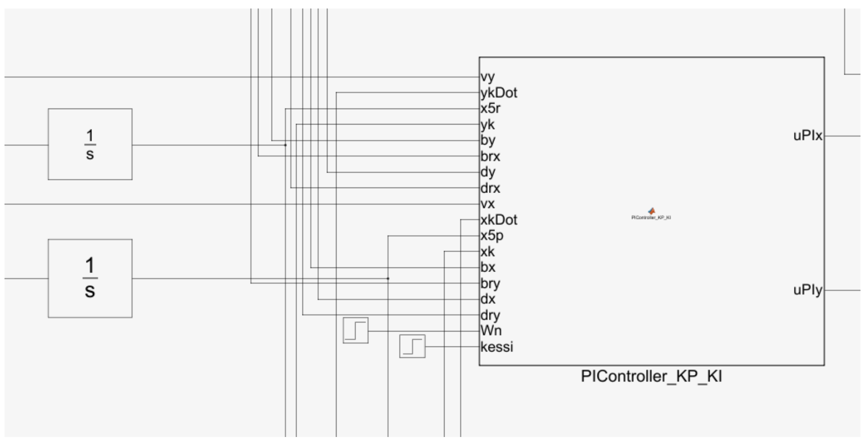
همه ی معادلات را وارد کرده و بلوک ها در صفحه ی بعد آورده شده اند.

بلوک LQR:

مقادیر Xkd , Ykd به ورودی مکانی مطلوب در سیمولینک به متلب داده شده است(مثل عه تابع پله)



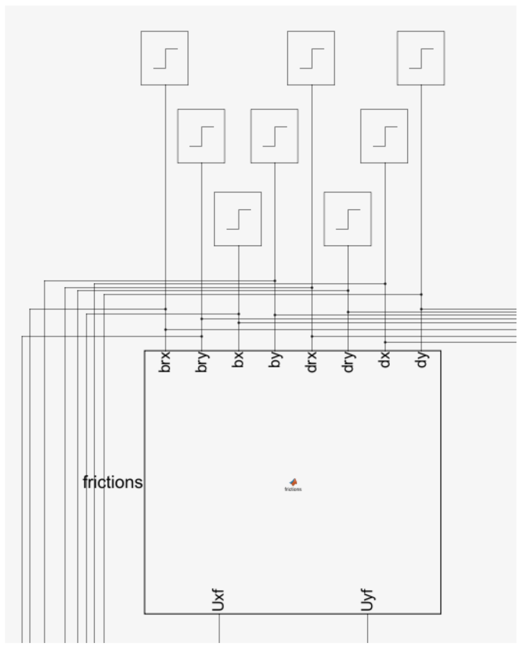
بلوک PI Controller:



بلوک FeedForward Compensation Term

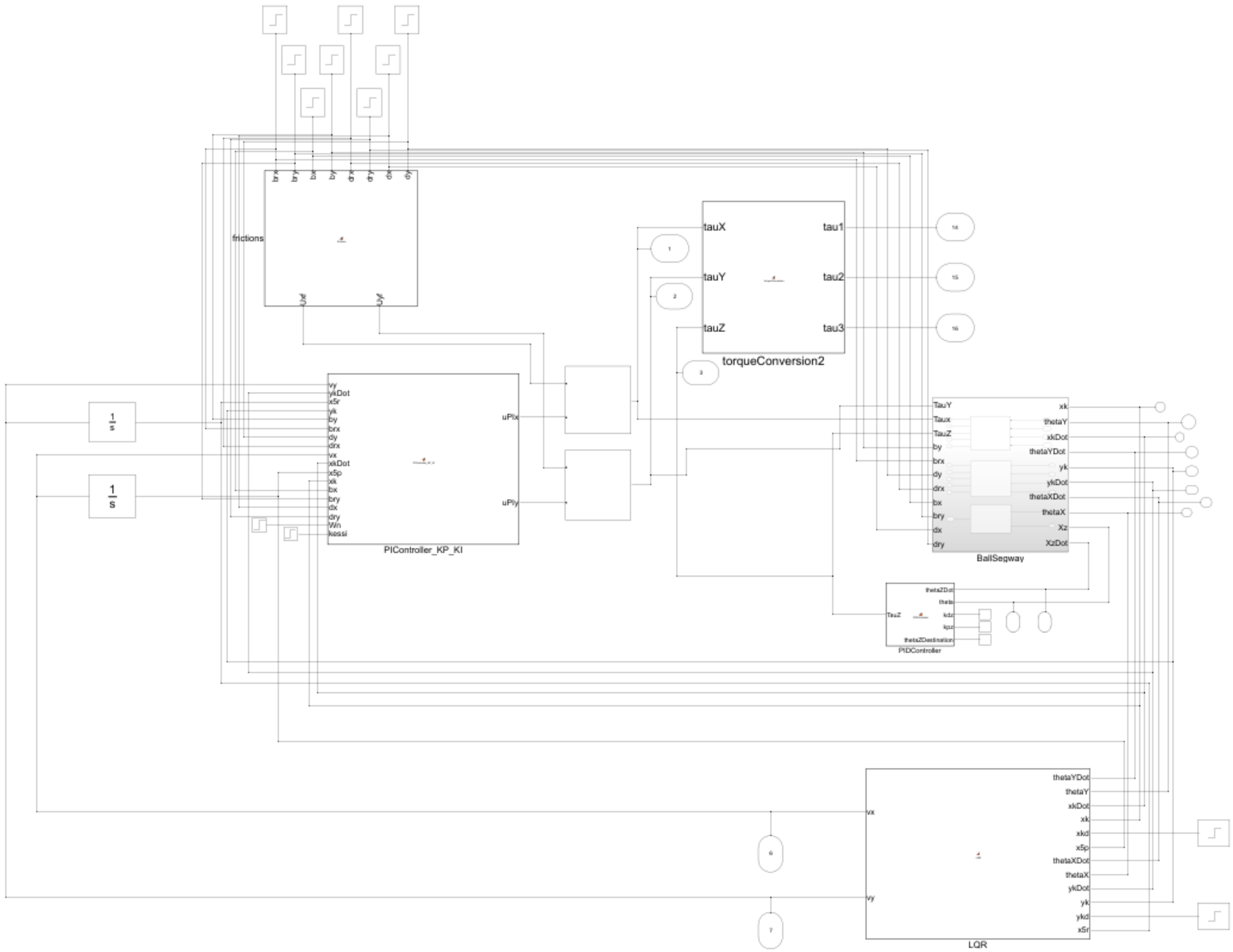
توجه: از آنجا که ضرایب اصطکاک در مقاله داده نشده اند این 8 ضریب اصطکاک را بنده به صورت ورودی برای واحد feedforwardCompensationTerm در نظر گرفتم.

به بلوک زیر توجه کنید:



نمودار کل بلوک دیاگرام پروژه:

این فایل به نام Allproject.slx در فایل آپلود شده موجود است.

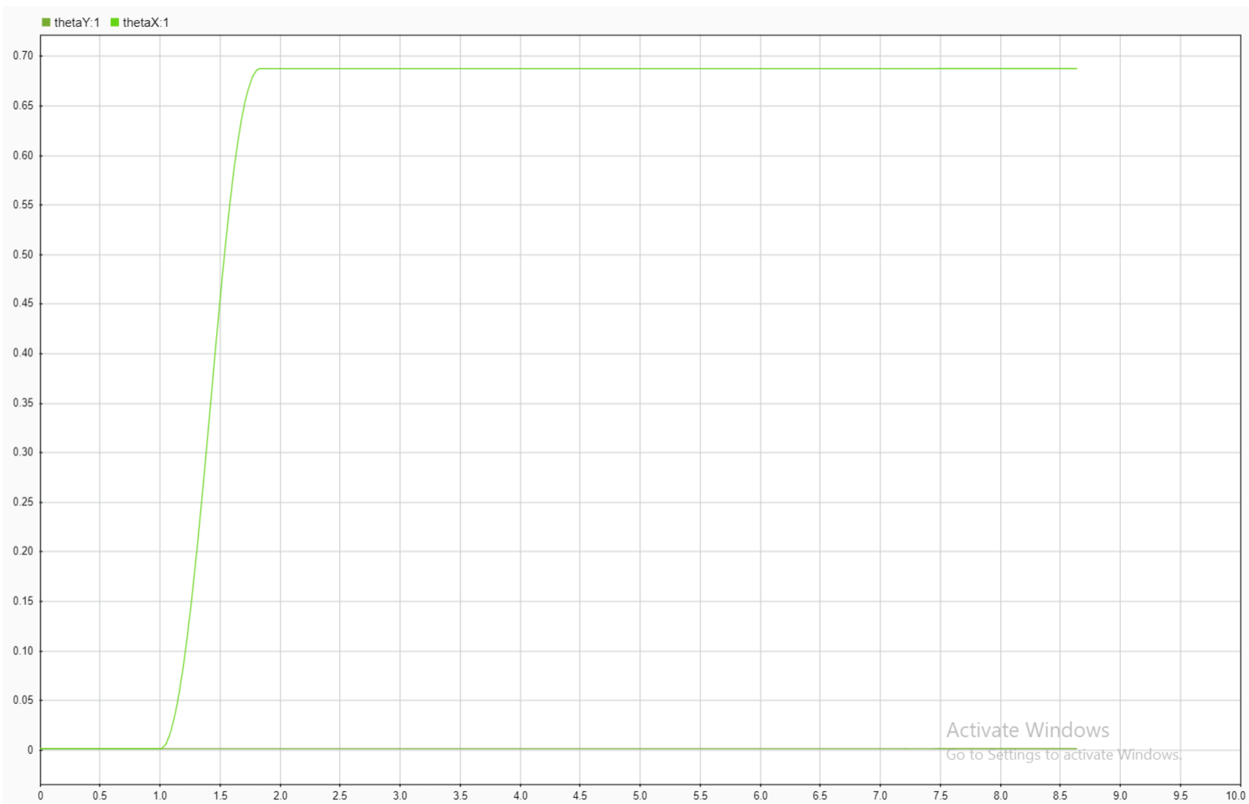


گام ششم:

متعادل کردن با یک زاویه اولیه غیر صفر:

تست لختی!!!

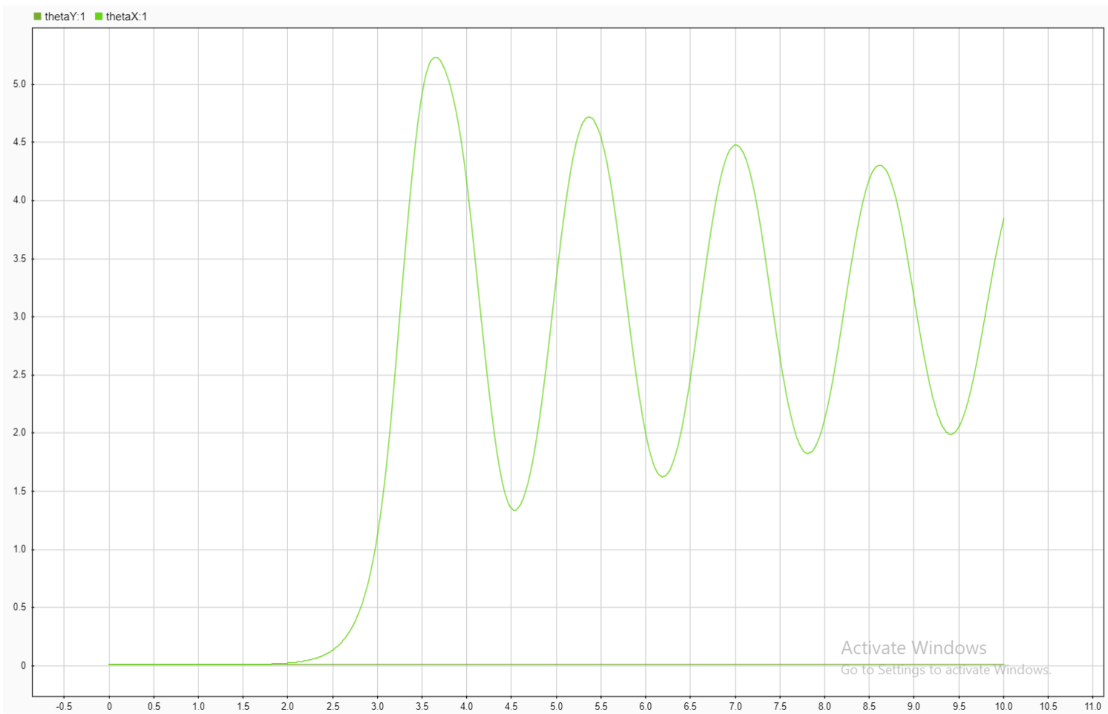
ابتدا تست لختی می گیریم که ایا کنترلر ما توانایی نگه داشتن خود در زاویه صفر را دارد یا خیر! با ضرایب معرفی شده و نکات ذکر شده در گزارش ما زتا را برار 0.2 و wn را برابر 2.5 در نظر میگیریم(برای کنترلر کننده یPI)(پارامتر های تعریف شده توسط خود متن) شکل به صورت زیر میشود:



همانطور که ملاحظه می کنید در تست لختی thetaY لختی خود را به خوبی حفظ می **کند اما thetaX ابتدا شروع می کند تا به حدودا 0.7 رادیان برسد و پس از آن به کندی و با نرخی بسیار ضعیف(به نظر می آید که ثابت شده است ولی چنین نیست)و رشد می کند(پس از ران کردن برنامه در بازه ی زمانی نسبتا زیاد دریافتیم که پس از گذشت مدت زمان بسیاااار زیادی(در حد 5000 ثانیه!) در تست لختی این زاویه به pi میل می کند :/** و این نشون می دهد که در حالتی که ما ورودی عی به سامانه نداده ایم کنترلر ما نتوانسته است اثر feedforwardCompensation را در راستای x برطرف کند(توجه کنید در حالتی که ورودی و حالت اولیه ای نداده ایم تنها خروجی های واحد friction است که مقدار غیر صفر به ballSegway اعمال می کند و همانطور که ملاحظه کردید با وجود اینکه تک تک مراحل طبق مقاله انجام شده است ولی داریم که در راستای X،زاویه ی ballSegway خود لختی خود را حفظ نمی کند.(حداقل اگر پس از مدتی زاویه به صفر می گرایید هم باز مطلوب بود اما طبق نمودار بالا مشخص است که این Damping رخ نمی دهد).

اگر به نمودار های Taux,TauY توجه کنیم می بینیم که سیر مثبت(اکیداصعودی) دارد و این یعنی که کنترلر دارد موافقت می کند با فاصله گرفتن از صفر.حال مقدار ضریب اصکاک را کم کرده تا زودتر نتیجه نهایی را مشاهده کنیم در این حالت خروجی به صورت زیر می شود:

در این حالت میبینیم که با اینکه تتای y به خوبی در حالت پایدار به نظر می ماند ولی کنترلر برای tx به نوعی دارد زاویه را خود به زاویه pi میبرد! در این حالت کنترلر به صورت شکل زیر میباشد:(**با کاهش زیاد اصطکاک** به حالت نهایی خیلی زودترسیدیم :/) در اصل ball زود لیز می خورد!(توجه کنید ما این کار را کردیم تا زود نقطه ای که به آن میل می کند را در بیاریم وگرنه که نباید ضریب اصطکاک را دستکاری کنیم)



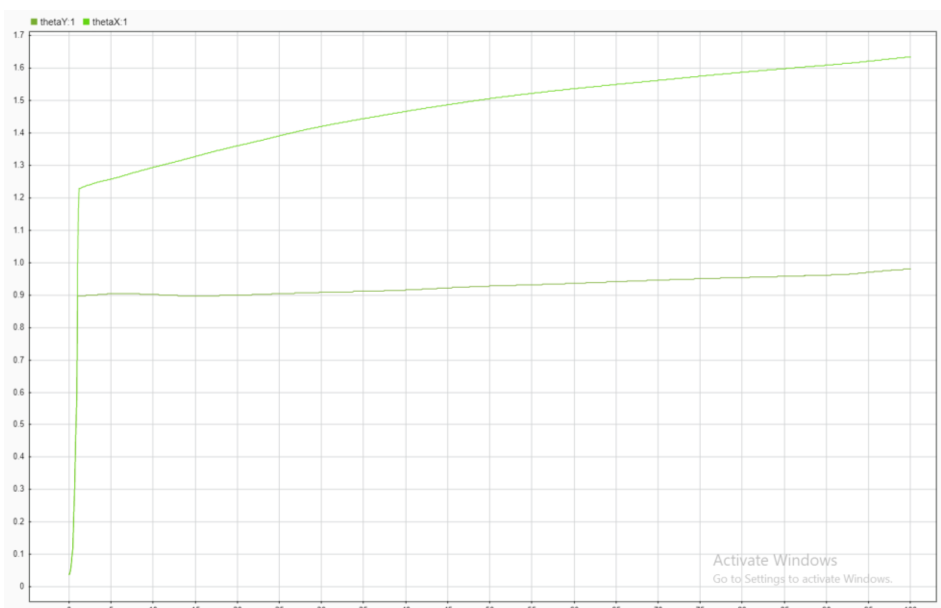
یعنی متاسفانه کنترل کننده خودش با همگرا شدن thetaX به pi موافقت می کند و ما این مساله را طبق ایمیلی به دکتر شرح دادیم و خانم دکتر به ما گفتند که در اولین قدم پس از بررسی صحت دینامیک اجزا به علامت ضرایب بپردازیم وخب ما هم از صحت روابط مطمعن شدیم و به بررسی تغییر علامت ها پرداختیم: اولین و ساده ترین حالت که به ذهن میرسد عوض کردن علامت ضرایب $k_{pp}, k_{ir}, k_{ip}, k_{pr}$ ببینیم میتوانیم که این مشکل را حل کرده یا نه، در این حالت جدول زیر را برای علامت ها و نتیجه پایداری یا ناپایدار کردن جواب داریم:(از آنجا که تست این بخش به شدت زمان بر بود من و دوستم آقای کلباسی به کمک یکدیگر نتایج این بخش را به دست آوردیم)

پایداری	kip	kir	kpr	kpp
ناپایدار	+	+	+	+

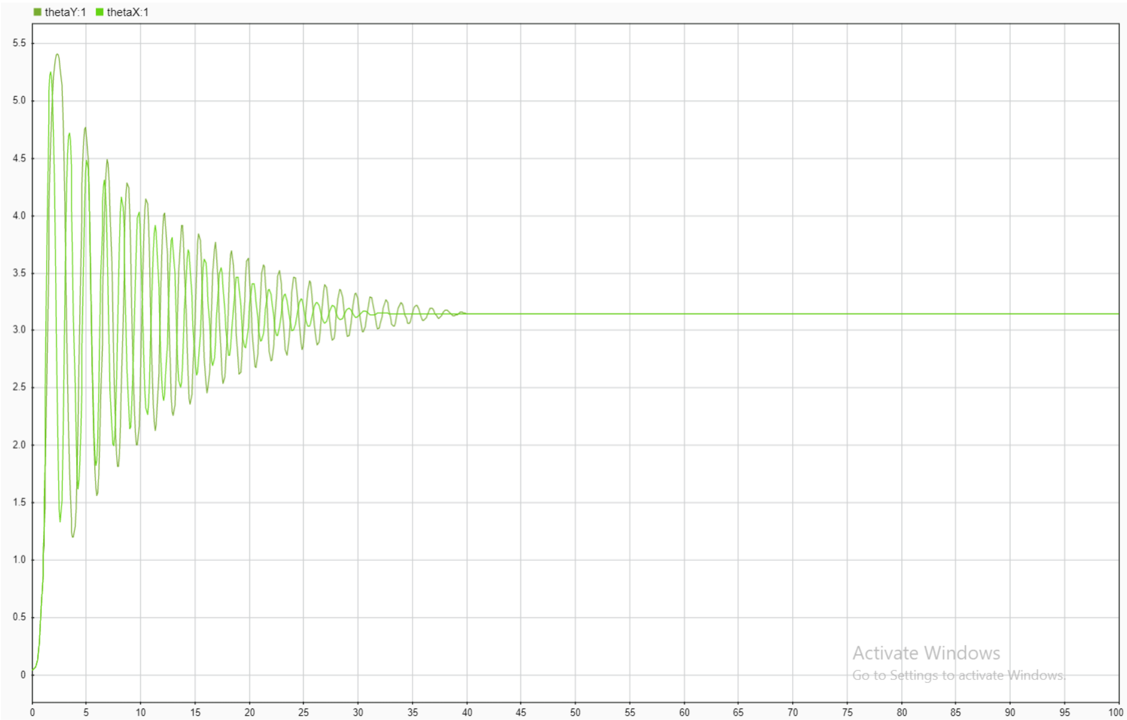
به شدت ناپایدار	+	+	−	+
پایدار همگرا(دیر یا زود) به π	−	+	+	+
ناپایدار	+	−	+	+
ناپایدار	−	+	_	+
ناپایدار	−	−	+	+
پایدار همگرا(دیر یا زود)به π	−	+	+	−
ناپایدار	−	−	−	−
ناپایدار	+	+	−	−
ناپایدار	+	+	+	−
به شدت ناپایدار	+	−	−	−
ناپایدار	+	−	−	+
ناپایدار	+	−	+	−
به شدت ناپایدار	−	+	−	−
ناپایدار	−	−	+	−

خب دیدیم که هر بلایی سر علامت کنترلر آوردیم نتیجه تغییر نکرد،حال برویم سراغ تغییر زتا و Wn بخش کنترلرPI (توجه کنید که این تست در حالتی که خروجی ما پایدار(هرچند به pi) انجام می شود، زاویه اولیه را طبق خواسته سوال روی دو درجه تنظیم میکنیم تا زود تر اثر انرا ببینیم.

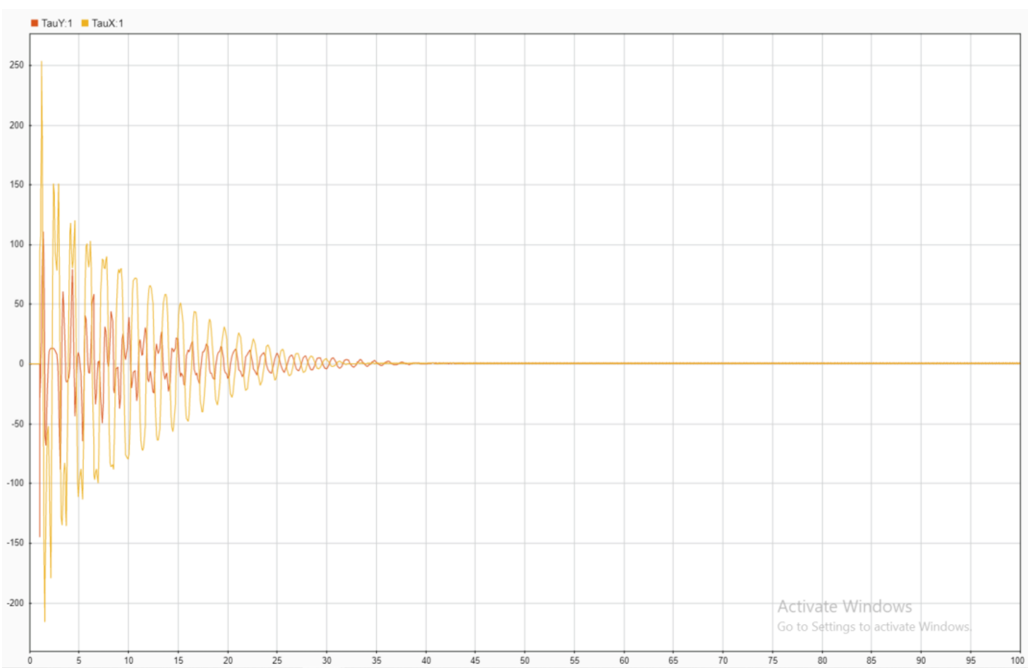
خروجی این بخش:(فعلا مقادیر زتا و Wn رو تغییر نداده ام):



همانطور که ملاحظه می کنید هم thetaX و هم thetaY با نرخ هرچند کم بهpi میل می کنن(کافیست بازه ی زمانی را بزرگ بگیریم) حال اگر ضریب اصطکاک را کاهش بدهیم داریم که خروجی به شرح زیر می گردد:(خروجی خیلی زود به pi می رسد)



خروجی کنترلر ها:(هم برای X و هم برای Y)



بررسی اثر تغییر زتا و Wn :

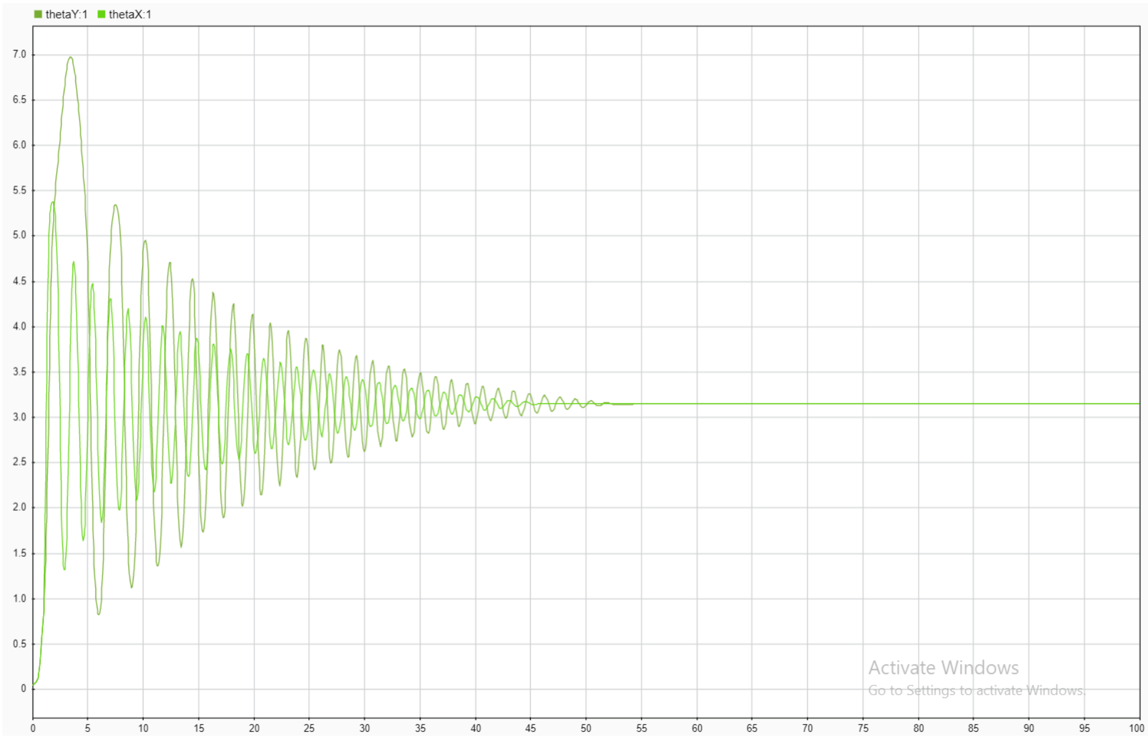
اثر تغییر Wn :

اگر مقدار wn و زتا را تغییر دهیم، باید دقت کنیم که wn را خیلی زیاد نکنیم وگرنه حاصل ناپایداری میدهد(مثلا به ازای Wn=10 داریم که خروجی به شرح زیر می گردد)

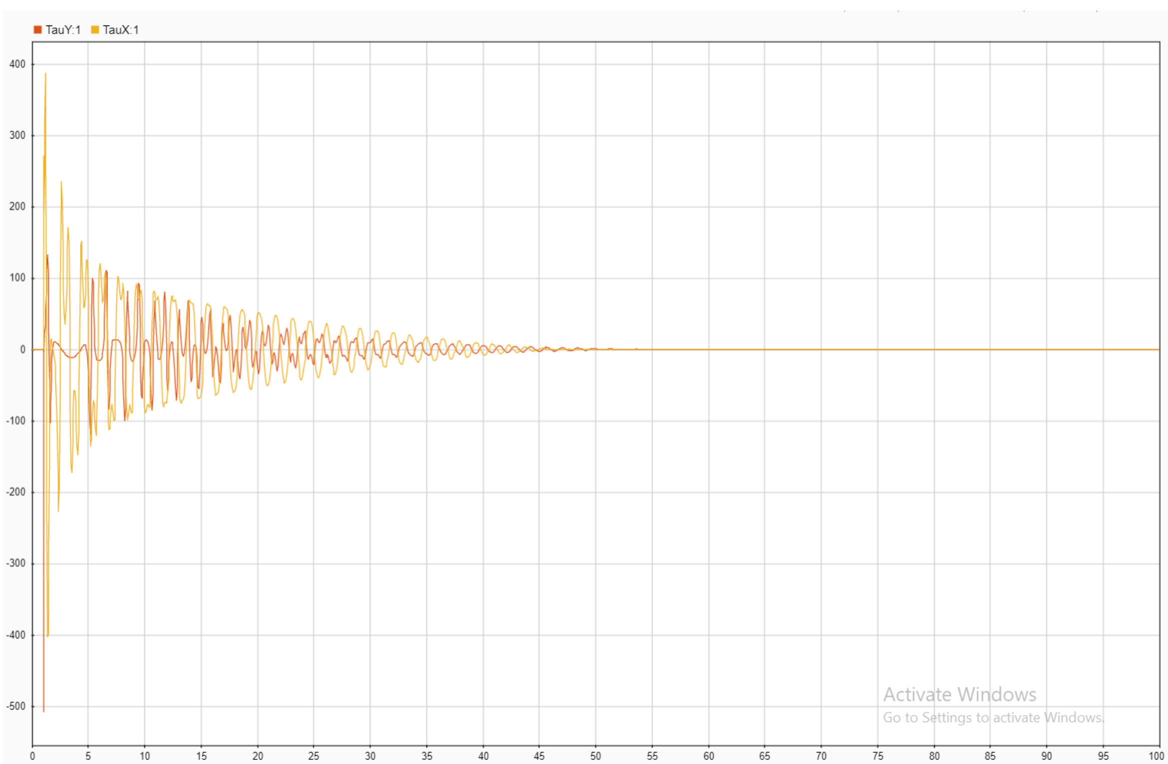
همانطورکه ملاحظه می کنید افزایش زیاد Wn سبب می شود که BallSegway عملکردش شدیدا مختل شده و هم هم thetaX و هم thetaY با افزایش Wn حول زاویه های بزرگی (مضارب بزرگتری از pi(2k+1) نوسان می کند)

(به حدود محور زمان توجه زیادی بکنید لطفا)

حال Wnرا به 5 افزایش می دهیم و خروجی به شرح زیر می شود:



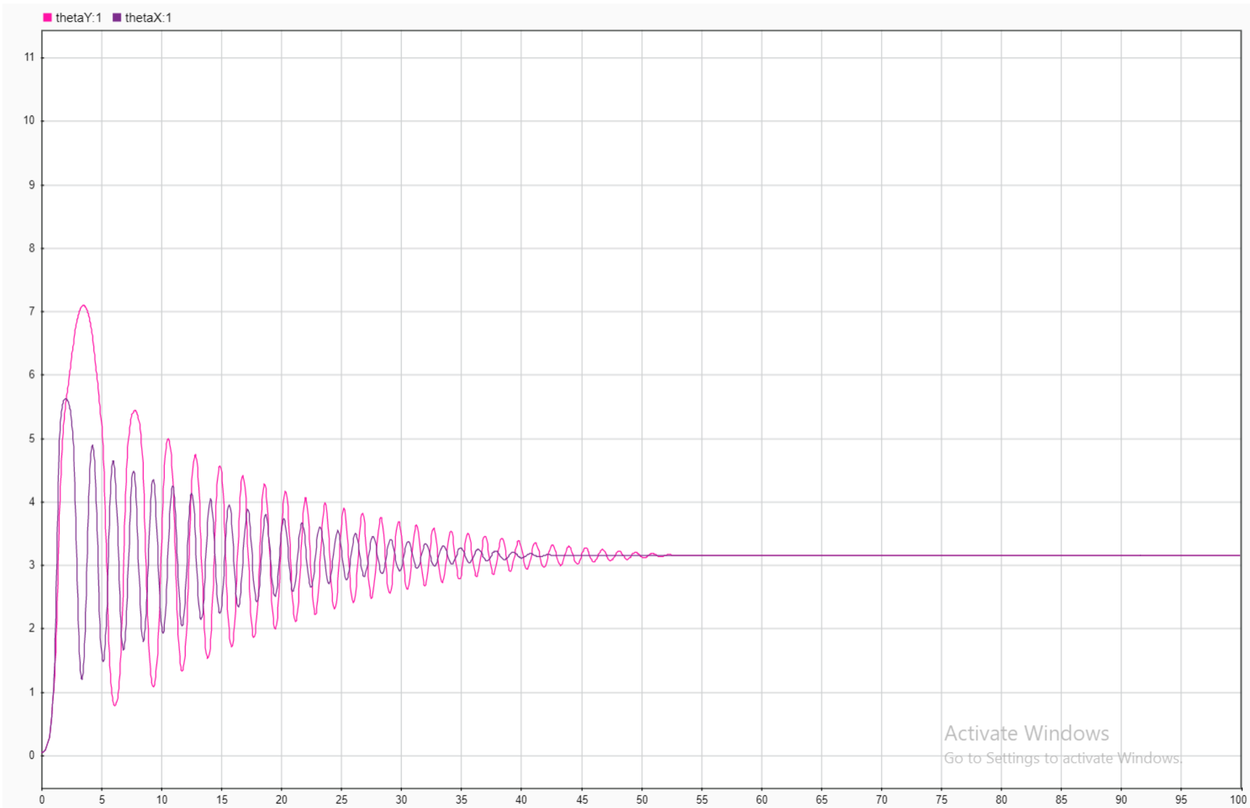
خروجی کنترلر ها در این قسمت:



همانطور که ملاحظه می کنید تغییر Wn صرفا در سرعت رسیدن به پایداری تاثیر دارد و تاثیری در درست شدن زاویه ندارد و هم چنان نمودارthetaY , thetaX حول pi(2k+1) نوسان می کنند(اگر Wn را زیاد کنیم از نوسان حول pi به 3pi و ... می رسیم نمونه مثالش در بالا آورده شده است).

اثر تغییر زتا:

اگر زتا را به 0.5 افزایش دهیم



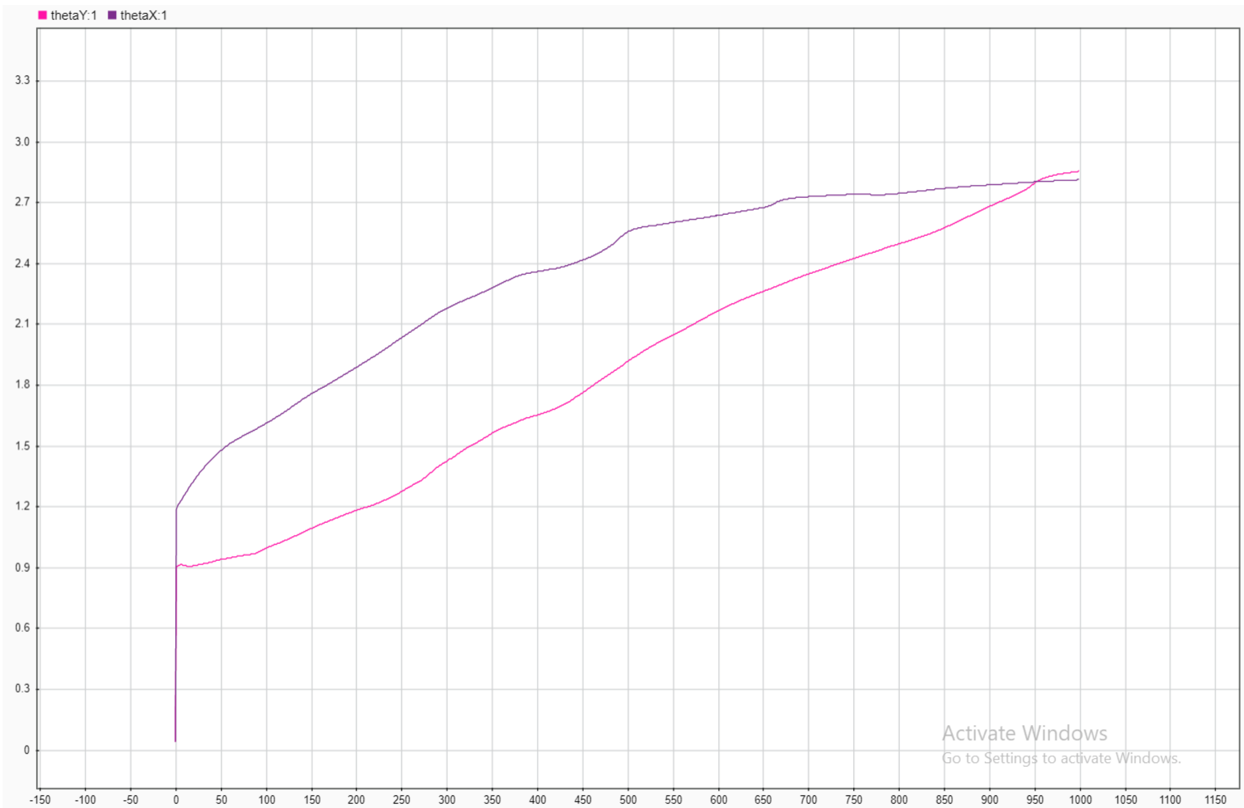
باز هم به صورت چشم گیر اثری را مشاهده نمیکنیم و تنها سرعت اسیلیت کردن کند تر و در نهایت دیر تر به 3.14 که همان زاویه π همگرا میشود.

پس ضرایب نیز مقادیرشان به درست کردن کنترلر کمکان نکردند.تا اینجای کار ما همه ی قسمت های موجود در مقاله را دستکاری کردیم و همانطور که مشاهده کردیم(و در ایمیل هم خدمتتون عرض کردم) نشد که نشد! حال تنها چیزی که می ماند ضریب اصطکاک ها اند! که خب اینا همگی مربوط به فیزیک مساله اند و فقط با زیاد کردن آن ها چون اصطکاک زیاد می شود صرفا گوی دیبر لیز می خورد.و در اصل کنترلر ما باید اونقدر قوی باشد که در محیط های مختلف عمل کند :///

در واقع **کنترلر از قصد به نظر در حالت تلاش به رساندن زاویه به pi** میباشد. ما این نکته را میدانیم که این مسئله به صورت مسئله اونگ معکوس درواقع می باشد برای همین نیز امری بدیهی است که زاویه π میل کند **ولی در ذات کنترلر که به ما مولفه هایش معرفی شده است قدرت نگه داشتن و مبارزه با این امر طبیعی نه تنها نمی باشد، بلکه از این امر به نظر استقبال میکند!**(همانطور که در ایمیل خدمتتون عرض کردم:/)اگر به حالت نیروی وارده در مرحله های قبل نگاه کنید **کنترلر کاملا در جهت رساندن زاویه به π عمل میکند و وقتی حدودا زاویه به π میرسد دیگر نیرویی وارد نمیکند**، این امر هم به راحتی قابل

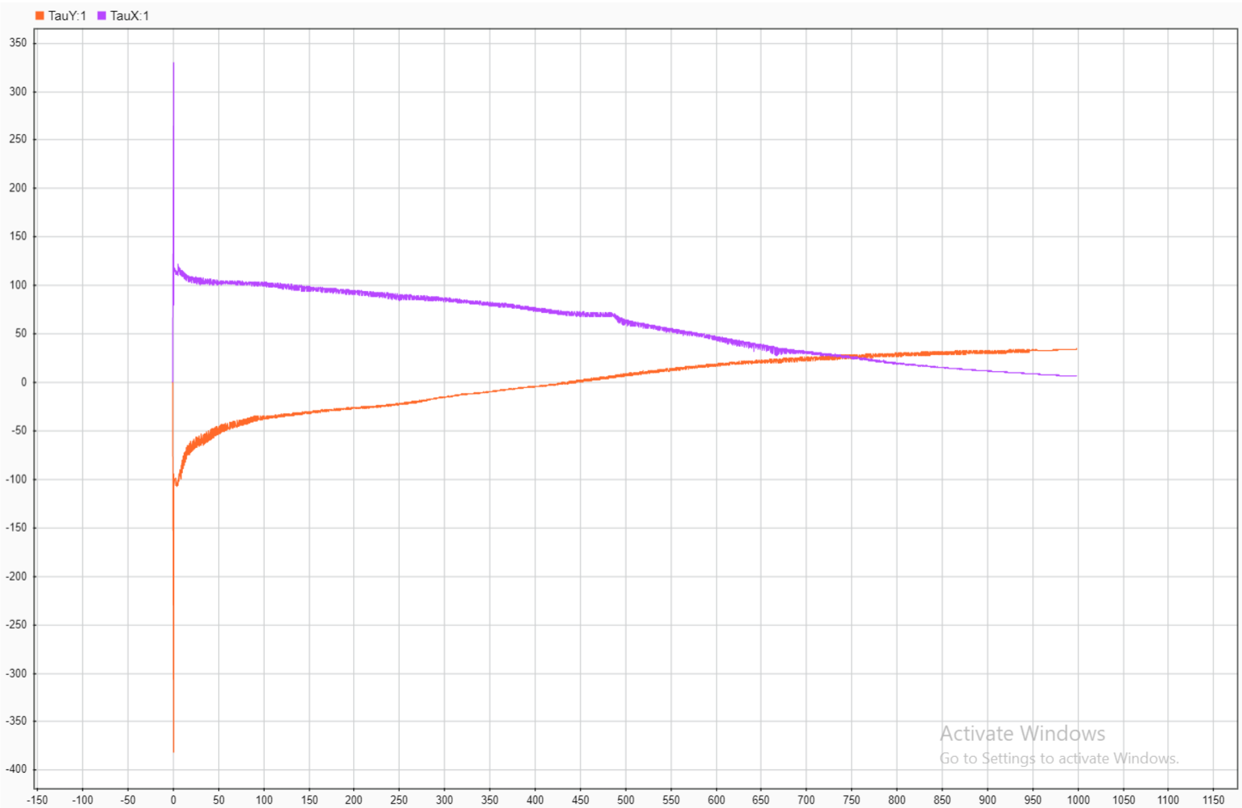
اصلاح نیست چون ما تمام حالت علامت های کنترلر و ضرایب مختلف ان و حتی ضرایب اصطکاک را امتحان کرده ایم پس به نظر نمیرسد این مشکل تنها با عوض کردن ضرایب رفع شود و **مشکل ذاتی** میباشد.

نمونه خروجی در حالتی که ضرایب اصطکاک افزایش می یابند:

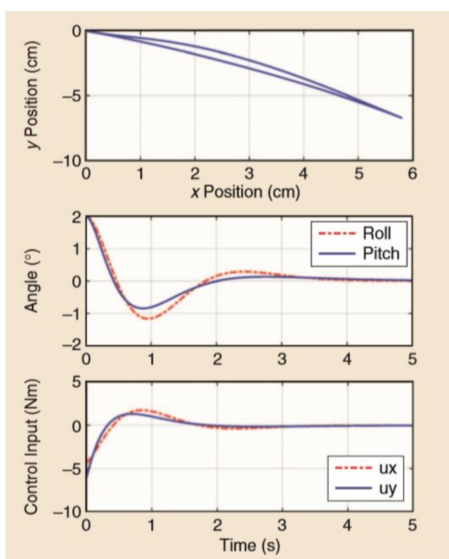


همانطور که ملاحظه می کنید صرفا با افزایش اصطکاک thetaX , thetaY صرفا دیرتر افزایش پیدا کرده و دیرتر به pi می رسند:/// **scale محور زمان توجه کنید و با قسمت های قبل مقایسه کنید.(توجه کنید که شرایط اولیه را من در ثانیه ی اول اعمال می کنم).**

نمودار کنترلر ها در این حالت:**<u>scale محور زمان توجه کنید و با قسمت های قبل مقایسه کنید</u>.(توجه کنید که شرایط اولیه را من در ثانیه ی اول اعمال می کنم).**



همانطور که ملاحظه کردید با اعمال هر تغییری(از Wn و زتا گرفته تا تک تک ضرایب کنترلر PI)به خروجی مطلوب نرسیدیم!و خروجی مطلوب طبق گفته ی مقاله در این حالت باید به شرح زیر باشد:



و توجیه آن نیز به این صورت است که ساختار کنترلر آنقدر باید قدرت داشته باشد تا زاویه های اولیه ی thetaX,thetaY را به صفر در زمان کوتاه بگرایاند.

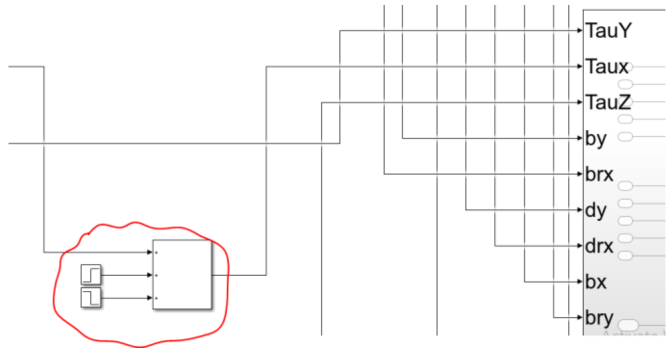
گام هفتم:

توجه کنید که ما ورودی را در ثانیه ی 1 می دهیم و ثانیه ی1 با ثانیه15 فرقی ندارد لذا ما اعمال ورودی های مدنظر را در ثانیه 1 ام می کنیم.

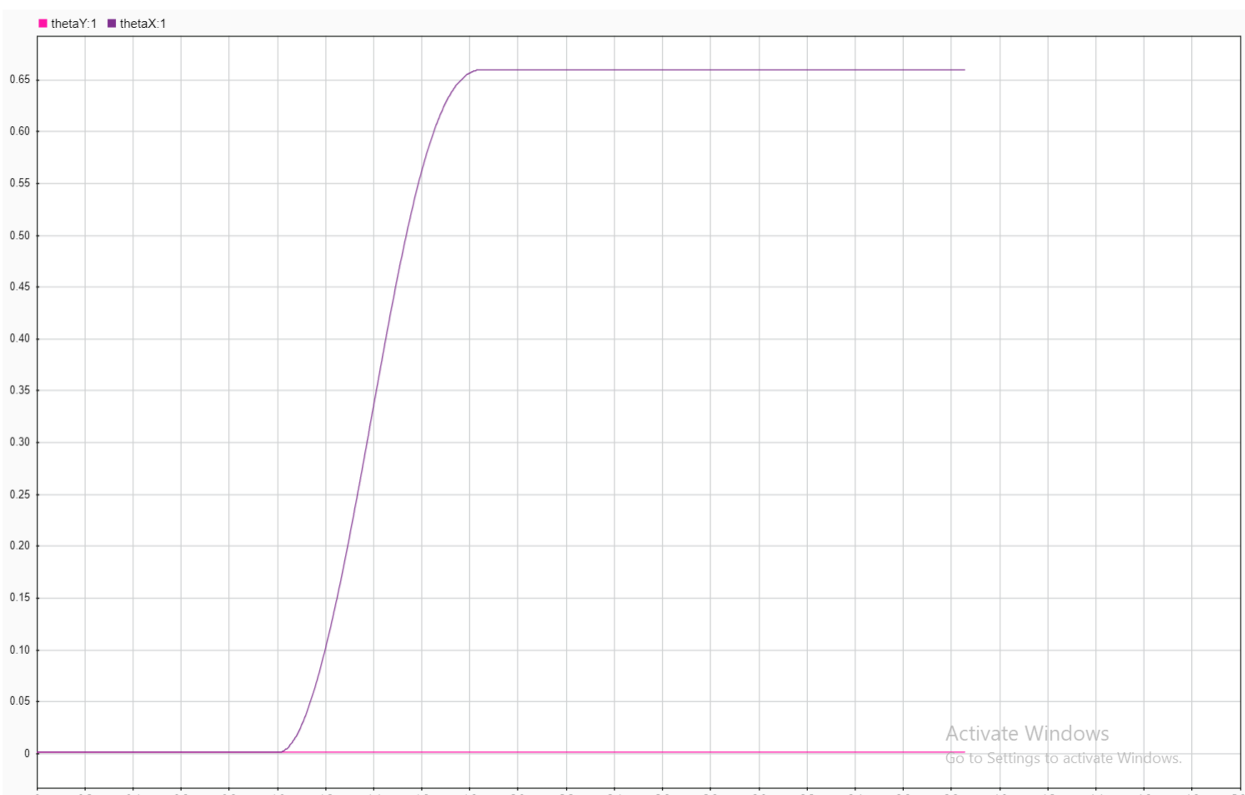
قسمت اول:

$$F = 50\ N\ , r = 0.22\ m \rightarrow TauX = r \times F \rightarrow TauX = 11\ N.m$$

حال برای شبیه سازی اعمال ضربه از طرح زیر استفاده می کنم که شبیه ساز اعمال ورودی ضربه در ثانیه ی 1 است.

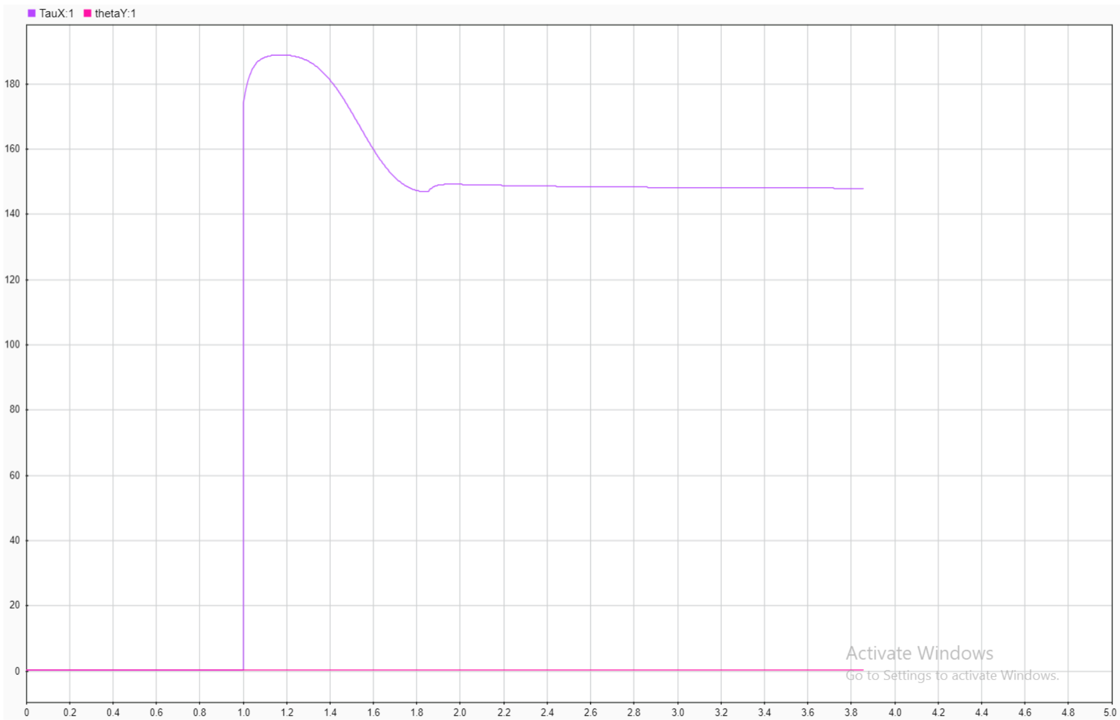


خروجی در این حالت عبارت است از:(thetaX,thetaY)



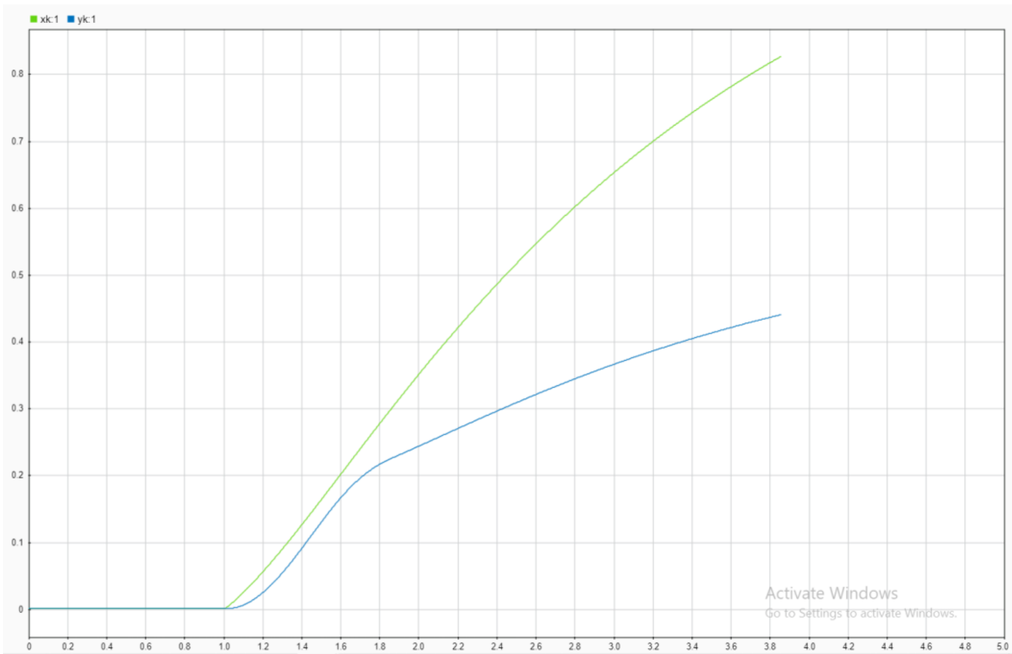
اعمال ورودی disturbance به صورت لحظه ی ای زاویه اولیه را تغییر می دهید و سپس خودش از بین می رود و طبق قسمت قبل داشتیم که اگر حالت اولیه مون غیر صفر باشه دینامیک مدار آن را به pi می‌گراياند و در زمان نسبتا زیاد دوباره thetaX به pi گراییده می شود.(**توجه کنید که مقدار thetaY صفر نیست و اگر در نمودار زوم کنیم می بینیم که این هم با مشتق ضعیفی اما در حال افزایش است و در نهایت پس از گذشت زمان نسبتا خوبی داریم که به pi گراییده می شود).**

نمودار قسمت کنترلر به شرح زیر است:

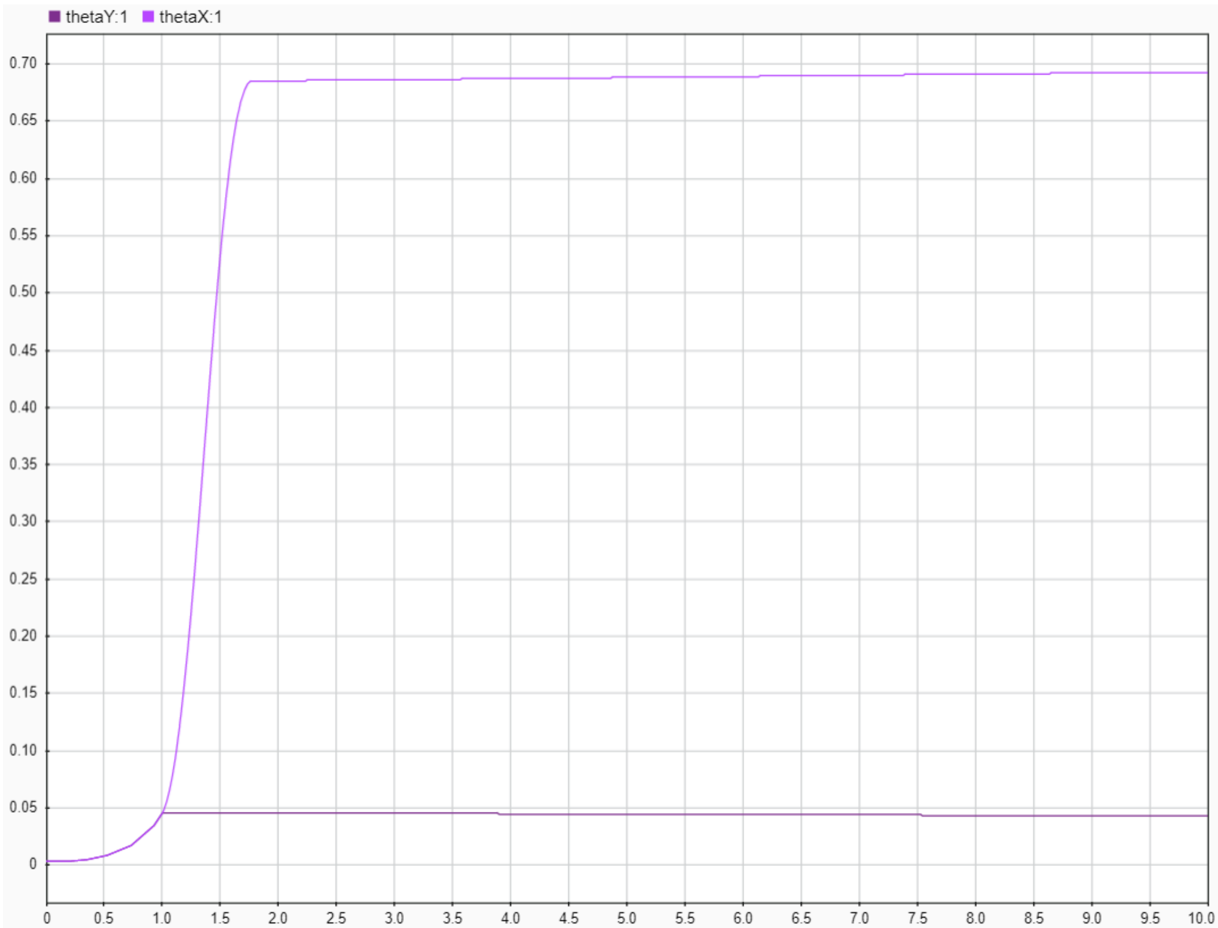


همانطور که ملاحظه می کنید کنترلر تا حدی تلاش کرده است که پس از اعمال ورودی با گشتاور تحمیلی وارد شده مبارزه کند و از لحاظ مبارزه با disturbance خارجی درست کار کرده است اما افسوس که به خاطر قضایای شرح داده شده در قسمت قبل نهایتا کنترلر thetaX,thetaY را به pi می گراياند.

نمودارهای xk,yk:

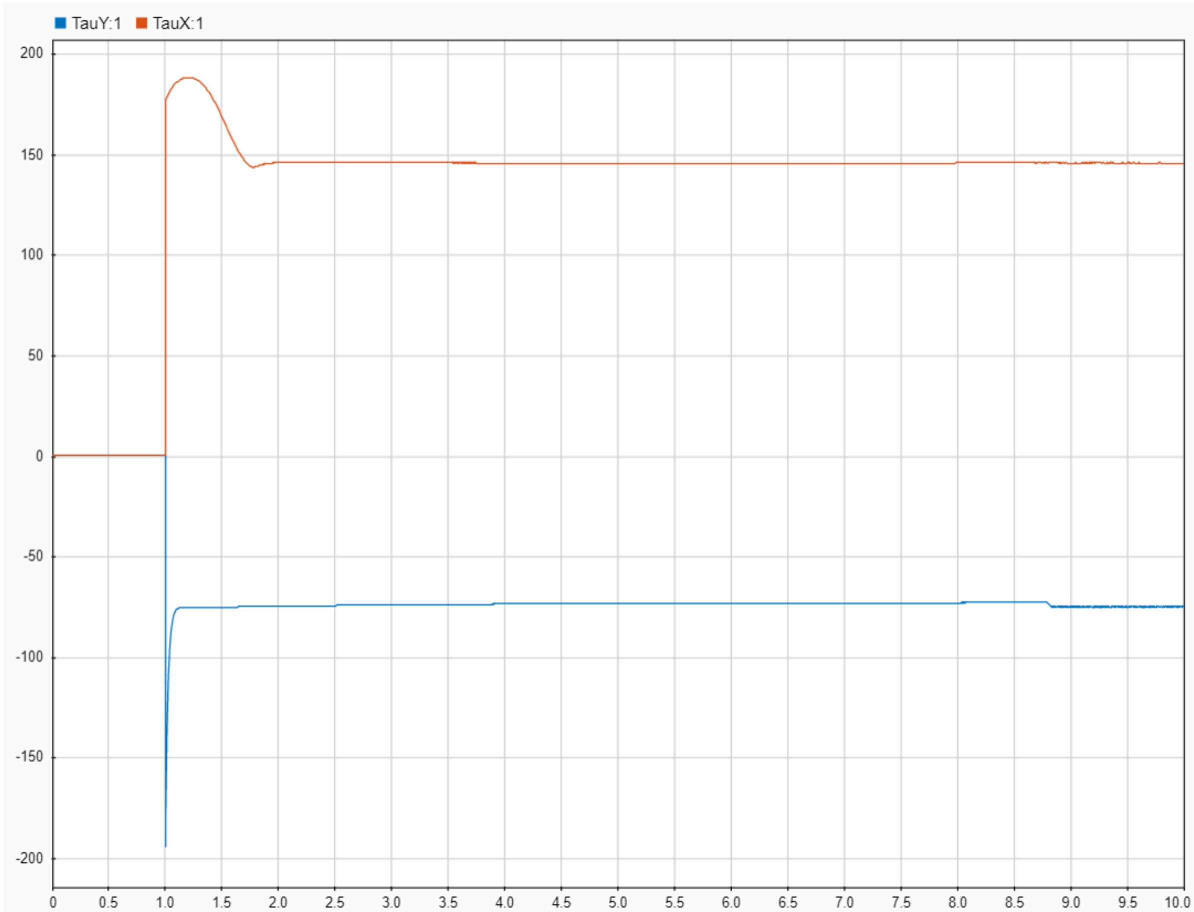


b) اعمال گشتاور به صفحه ی x-z:



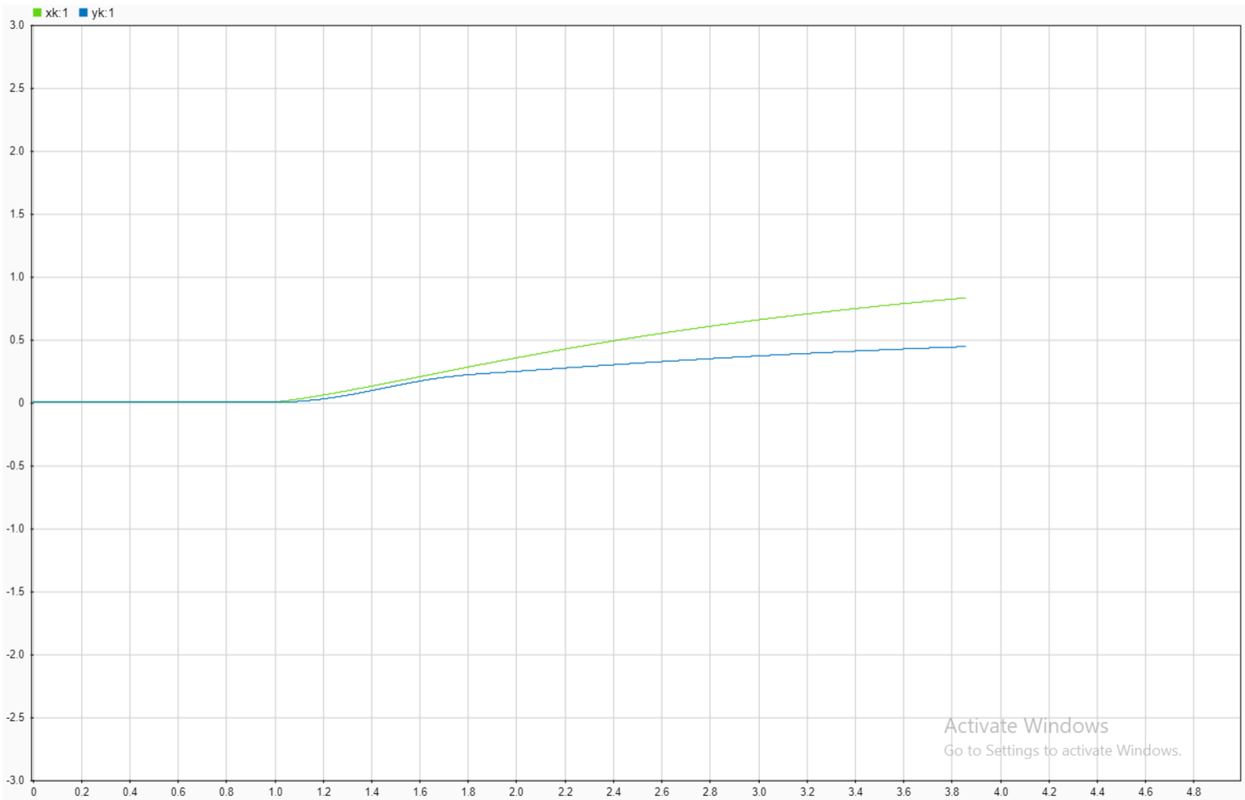
اعمال ورودی disturbance به صورت لحظه ی ای زاویه اولیه را تغییر می دهید و سپس خودش از بین می رود و طبق قسمت قبل داشتیم که اگر حالت اولیه مون غیر صفر باشه دینامیک مدار آن را به pi می گرایاند و در زمان نسبتا زیاد دوباره thetaX به pi گراییده می شود.(**توجه کنید که مقدار thetaY صفر نیست و اگر در نمودار زوم کنیم می بینیم که این هم با مشتق ضعیفی اما در حال افزایش است و در نهایت پس از گذشت زمان نسبتا خوبی داریم که به pi گراییده می شود).**

نمودار قسمت کنترلر به شرح زیر است:



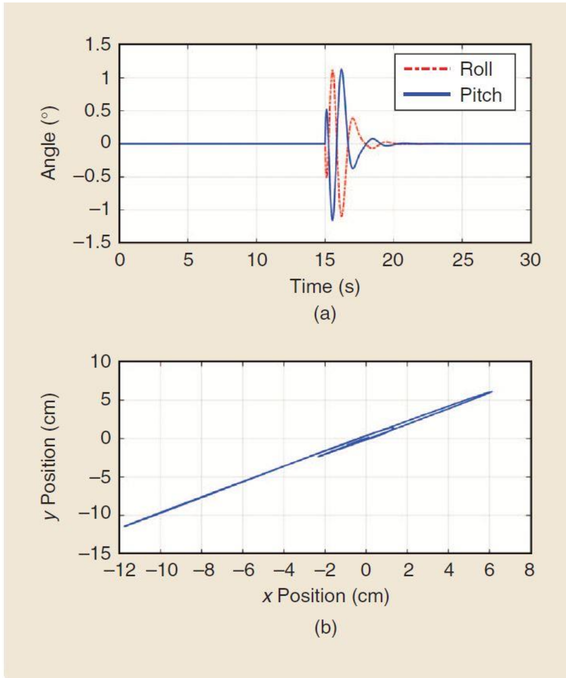
همانطور که ملاحظه می کنید کنترلر تا حدی تلاش کرده است که پس از اعمال ورودی با گشتاور تحمیلی وارد شده مبارزه کند و از لحاظ مبارزه با disturbance خارجی درست کار کرده است اما افسوس که به خاطر قضایای شرح داده شده در قسمت قبل نهایتا کنترلر thetaX,thetaY را به pi می گرایاند.

نمودار های Xk,Yk:



c)مقایسه و رسم نمودار های xk,yk در هر قسمت رسم و گزارش داده است حال برویم سراغ درست قضیه و مقایسه با نتایج موجود در خود مقاله.

نتایج درست مقاله:



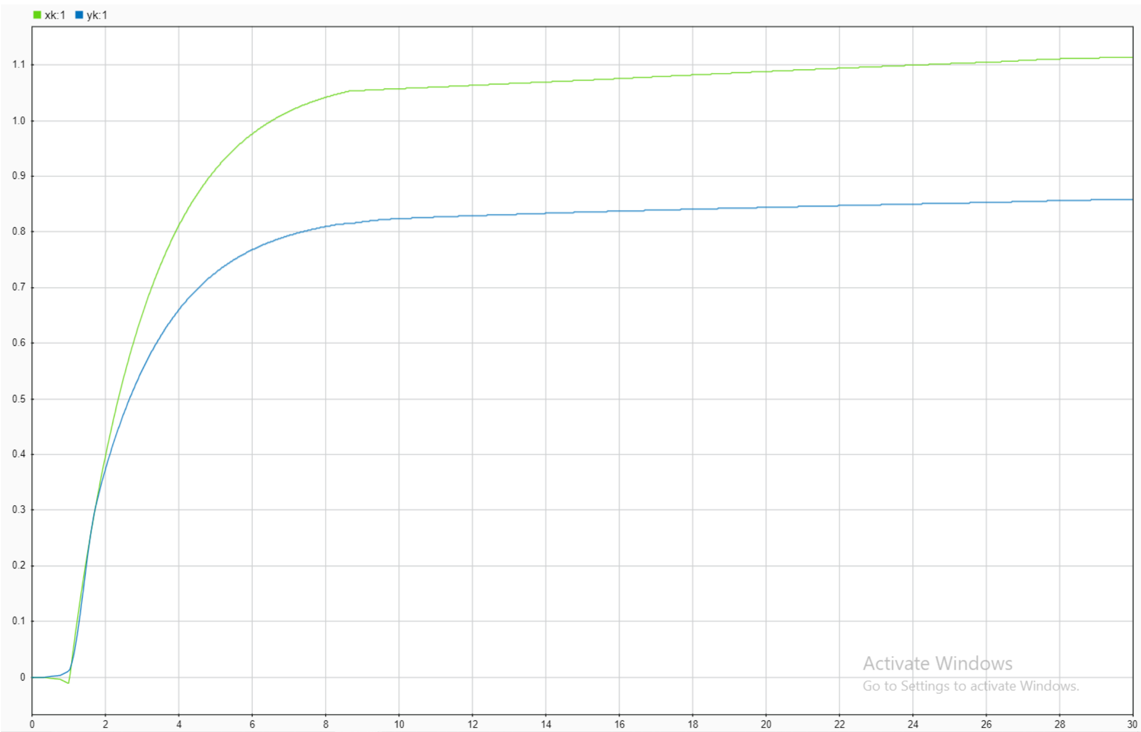
مقایسه و استنتاج:

همانطور که ملاحظه می کنید نتیجه ی مشترک بین مدل سازی ما و مقاله در این است که هر هردوحالت واحد کنترلر با تنظیم گشتاور خروجی به شدت با عامل disturbance مقابله می کند(همانطور که در نمودار ها رسم شده است)

گام هشتم: انتقال به مکان مورد نظر:

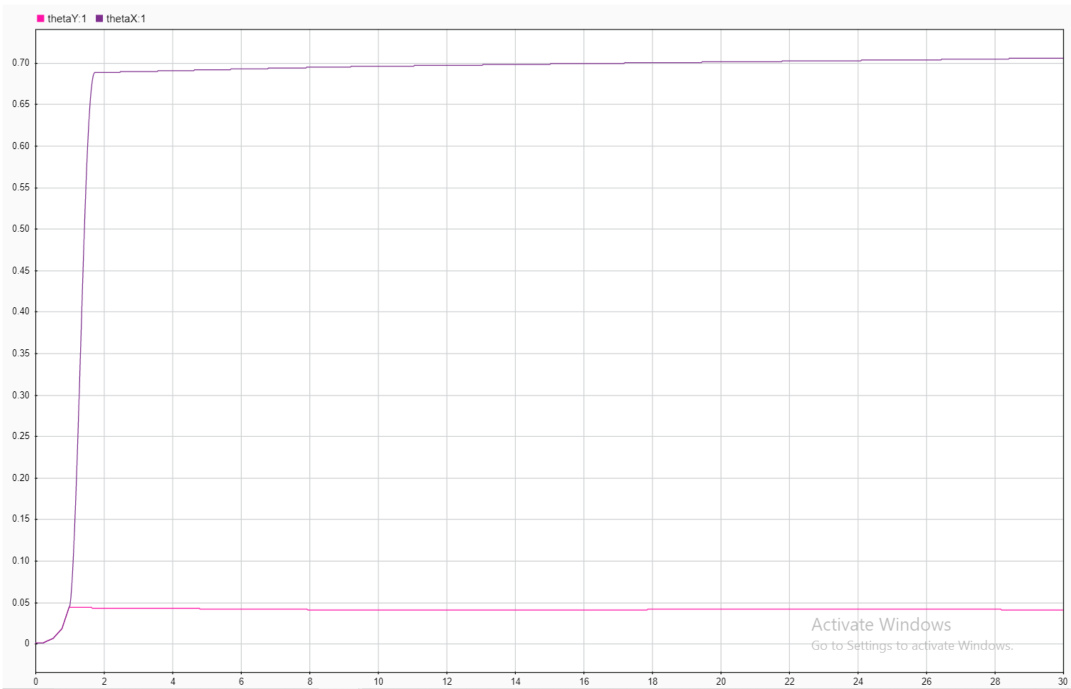
a) یک طراحی درست برای کنترلر حالتی است که خطای ماندگار به ورودی پله نداشته باشیم و در محاسبات طراحی کنترلر ما نیز این باید رعایت شده باشد، پس طبق انتظار ورودی پله ما همان نقطه ای است که می خواهیم به ان برود با فرض اینکه کنترلر توانایی ردیابی مرجع را به دقت کامل دارد.حال برویم سراغ تست مدل:

نمودار های Xk,Yk:



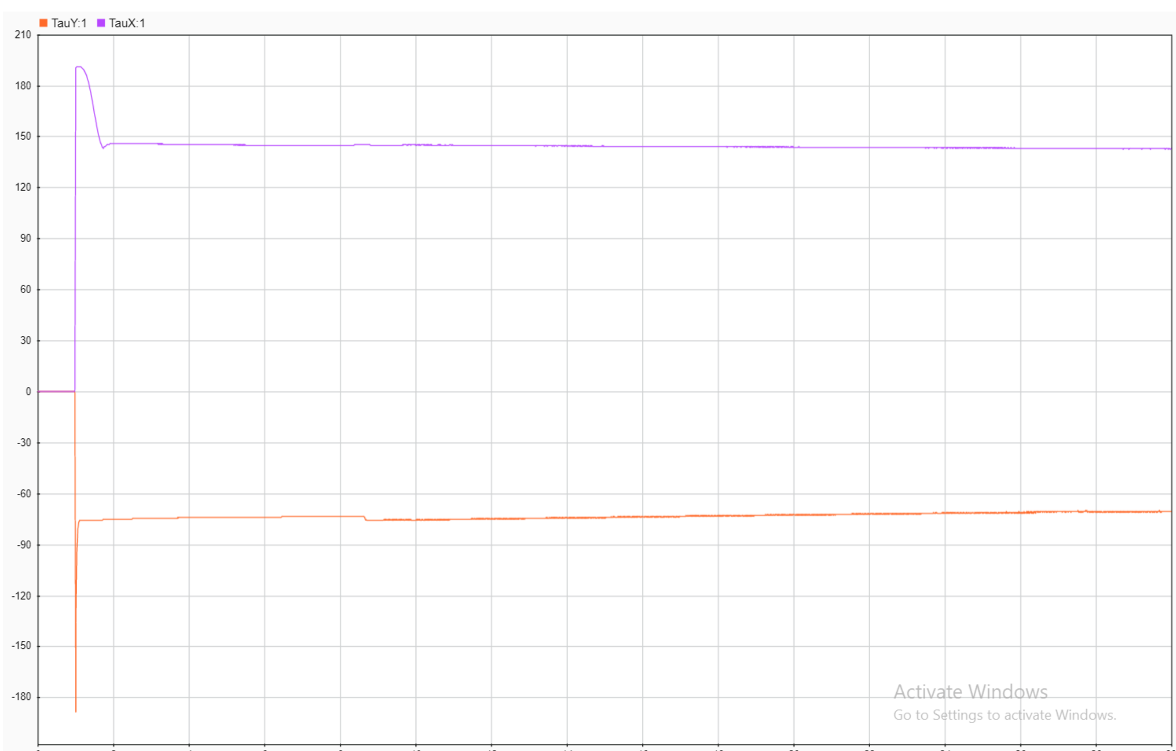
همانطورکه ملاحظه می کنید هرچند با نرخ کم ولی xk,yk ناپایدار رفتار می کنند ولی با شیب ضعیفی واگرا می شوند ://

نمودار های thetaX,thetaY:



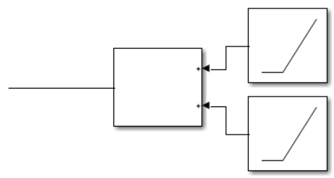
که همانند همیشه پس از تاثیر پذیری از ورودی داده شده در ثانیه 1،با مشتقی بسیارناچیز اما موجود به pi می رود:/// thetaX زودتر از thetaY به pi می رود.

نمودار سیگنال های کنترلی:



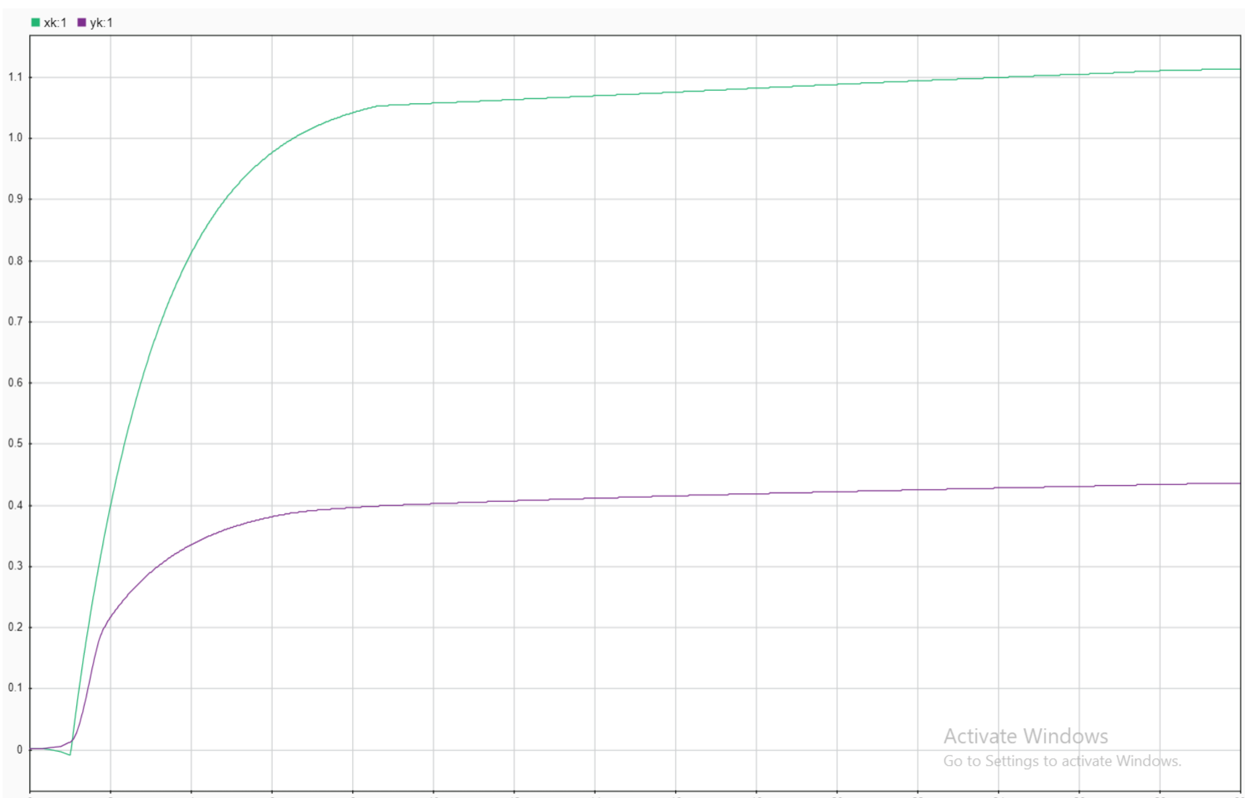
همانطور که ملاحظه می کنید کنترلر تا حد خوبی تلاش کرده است که پس از اعمال ورودی با گشتاورناگهانی تحمیلی وارد شده به علت تغییر شدید ورودی به خاطر Step عی بودنش مبارزه می کند و تا حدی این تغییر شدید را اصلاح می کند اما افسوس که به خاطر قضایای شرح داده شده در قسمت قبل نهایتا کنترلر thetaX,thetaY را به pi می گرایاند.

b) برای ورودی شیب باید شیب را به اینگونه بسازیم که دو شیب را از هم با یک فاصله کم کنیم که انگار با شیب واحد به مقدار نهایی میرسد به کمک باکس زیر:



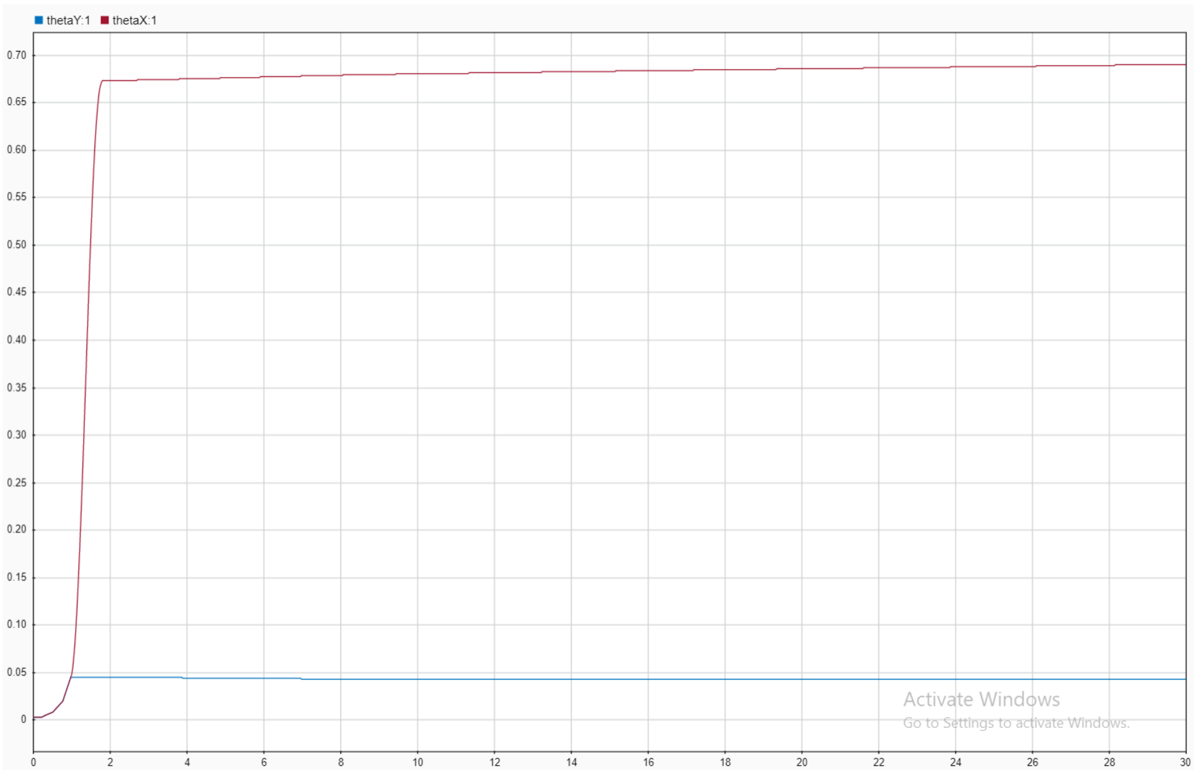
توجه کنید که علامت شیب ها وارون یکدیگر اند.

سیگنال های x,y به شرح زیر اند:



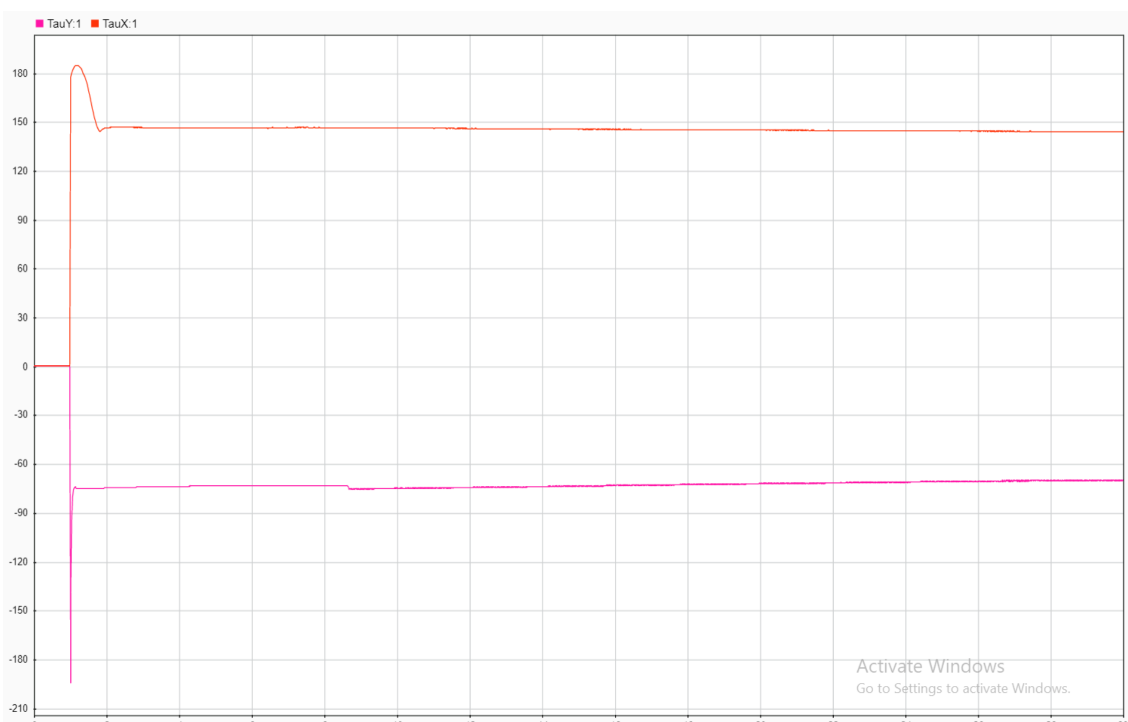
اگر به محور افقی دقت کنید متوجه میشوید کمی ارام تر سرعت تغییرات شده است.

نمودار های thetaX,thetaY:

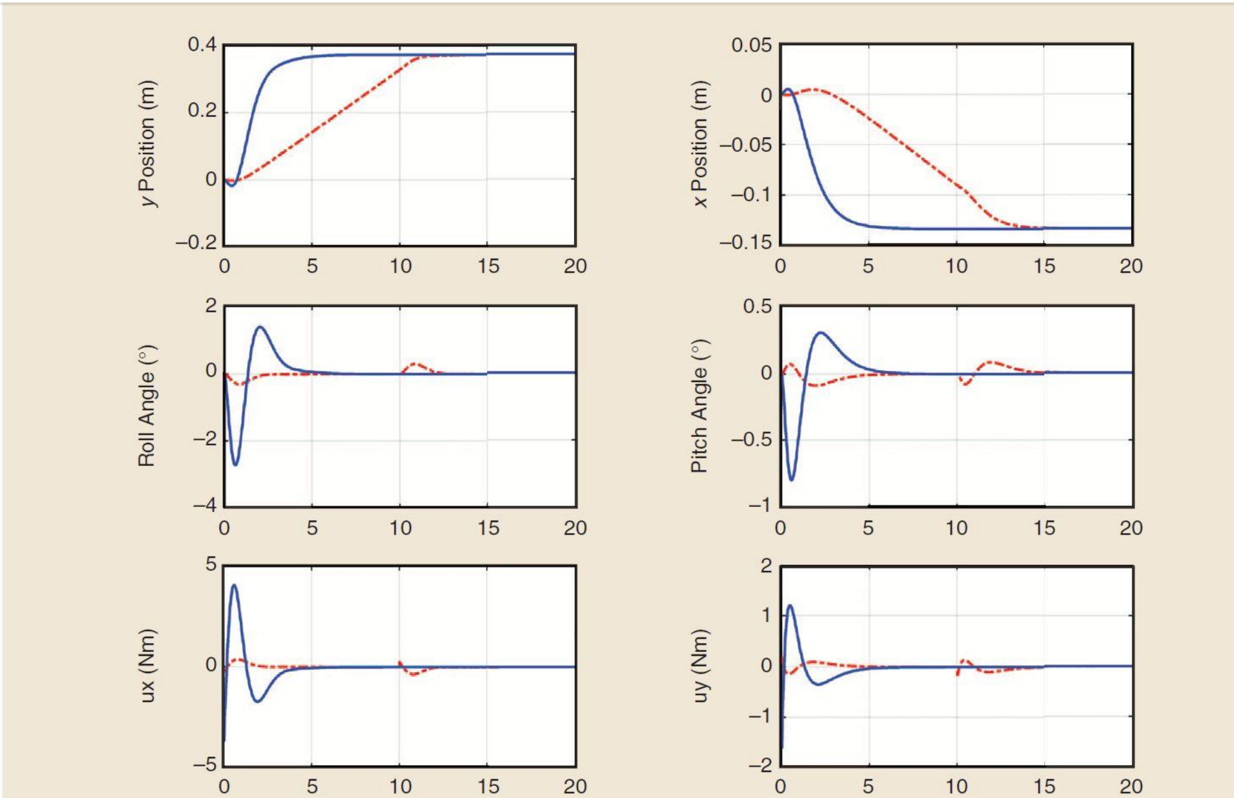


همانطور که ملاحظه می کنید در مقایسه با قسمت قبل داریم که اندکی نرم تر و با سرعت کمتر thetaX,thetaY افزایش می یابد.

خروجی کنترلر ها:



مقایسه با جواب های شبیه سازی: در شبیه سازی ورودی پله باشد به صورت شکل زیر انجام شده است:



در جواب های مقاله ردیابی مرجع به خوبی انجام شده است و کنترلر ان توانای هندل کردن تغییر شدید زاویه را دارد اما همانطور که ملاحظه کردید این توانایی خیلی در کنترلر و مدل سازی ما موجود نبود.

c) توجیه خروجی های مقاله (داریم که خروجی های کنترلر و مدل سازی ما آن چنان مطلوب نبود)

در مقایسه پله با شیب میتوان این را گفت که سرعت پله بیشتر میباشد(هرچند در ابتدا مقدار خارج از مسیر حرکت میکند) ولی انچه که در شیب بهتر میباشد تغییرات زاویه می باشد، **در ورودی شیب زاویه ها به نسبت خیلی کم تغییر میکند** ولی ورودی پله مانند **ضربه ای به سیستم عمل** میکند و پس پاسخ به ورودی شیب نرم تر از پاسخ به ورودی ضربه است.

در ورودی پله سرعت ردیابی پاسخ زیاد است ولی تغییرات ناخواسته ی زاویه به خاطر شیب شدید آن زیاد است اما در ورودی شیب سرعت ردیابی کمتر است ولی زاویه آرام تر است و لذا سامانه را خیلی کمتر از حالت ضربه تهدید به ناپایدار شدن می کند.

جمع بندی:

ما برای زاویه z توانستیم به خوبی انرا کنترل کنیم و کنترل چرخشی خوبی روی ان داشته ایم ولی متاسفانه به کنترل حرکتی و تعادلی خوبی دست نیافتیم اما همانطور که در قسمت های قبل ملاحظه کردید کنترلر ما ورود گشتاور های تحمیلی ناگهانی را متوجه می شود و با تولید گشتاور وارون سعی در اصلاح و مقابله با به هم ریختن شرایط می گردد ولی خب چه کنیم که هرچه تست و آزمایش کردیم(اعم از تغییر هر ضریب و هر معادله و هر علامتی!!) کنترلر به حالت سالم خود نرسید.